

UNIVERSIDAD CARLOS III DE MADRID
ESCUELA POLITÉCNICA SUPERIOR
INGENIERÍA TÉCNICA EN INFORMÁTICA DE GESTIÓN



PROYECTO FIN DE CARRERA

*APLICACIÓN DE COMERCIO ELECTRÓNICO PARA PEQUEÑAS Y MEDIANAS EMPRESAS
A TRAVÉS DE LAS TECNOLOGÍAS OPEN SOURCE*

AUTOR: JAVIER ARROYO MORENO

TUTOR: JOSE LUIS MORATO LARA

Agradecimientos

A mis padres, que gracias a su ayuda y apoyo he logrado ir superando los distintos obstáculos y retos que van surgiendo y llegar hasta este momento en el que se cierra otra etapa.

A mi novia Blanca, gracias por aguantarme y apoyarme todos estos años y, sobre todo, estos últimos meses.

A Juanjo, que me ayudó, orientó y animó enormemente en los primeros pasos en el mundo laboral en empresa y de la que aprendí un espíritu de trabajo y de aprendizaje continuo que intento plasmar en mi día a día.

A Carlos, ya que no ha sido fácil compaginar nuestro trabajo con la elaboración de este proyecto.

Al CAU de Getafe, del que tan buenos recuerdos conservo por lo rápidamente que me acogieron y enseñaron y el gran personal que lo compone.

Al departamento de Biblioteconomía y Documentación de Getafe, donde empecé mis primeros pasos.

A mi tía M^a Jesús, gran apoyo y ayuda en mis días de universidad.

A mi tutor, Jorge, el cual ha aguantado mis numerosos cambios de proyecto y mi falta de tiempo para terminarlo.

Andromedical, primera parada en el mundo laboral y donde me entró el gusanillo por el mundo E-commerce.

Por último, quiero agradecer a todas aquellas personas que, de una forma u otra, me han ayudado a llegar hasta este punto.

Resumen

Nos encontramos ante la necesidad de que pequeños empresarios tengan la posibilidad de ofrecer sus productos en todo el mundo pero con las limitaciones económicas inherentes a la condición de “pequeño empresario”. Además, actualmente existen grandes plataformas de venta en el mercado que ofrecen gran cantidad y variedad de productos, con lo que debemos de buscar la forma de diferenciarnos.

Podemos encontrar numerosas soluciones, pero debemos asegurarnos de implementar aquella que mejor cumpla los requisitos y condiciones, que nos dé cierta libertad para diferenciarnos en la forma en la que llegamos al cliente final y que tenga la mejor perspectiva de futuro.

Buscamos potenciar el desarrollo del comercio electrónico en la pequeña y mediana empresa como forma de crecimiento y diversificación de la misma, permitiendo la distribución de sus productos o servicios por todo el mundo sin necesidad de desplegar un sistema logístico y empresarial propio ni la inversión de ingentes cantidades de dinero para la creación de una red de ventas.

Para el desarrollo de dicho sistema se abordará el problema desde dos perspectivas: la plataforma de venta y gestión y el sistema hardware que hará posible su funcionamiento. Éste último suele estar muy presente en el proyecto si hablamos de grandes empresas, pero en el escenario en el que se enfoca este proyecto, pequeña y mediana empresa, suele ser el gran olvidado, generando problemas a medio y largo plazo.

Para la elección del sistema de venta y gestión buscaremos la plataforma que permita aunar potencia con fiabilidad, personalidad y coste limitado. Para el sistema hardware nos decantaremos por sistemas Cloud que nos permitan la máxima flexibilidad y escalado de recursos en tiempo real así como una gestión completa del mismo que posibilite el ahorro de recursos y costes.

Como resultado obtenemos una plataforma que nos permite competir con grandes distribuidores ya afianzados en el mercado por la capacidad de personalizar nuestro producto y diseño acorde con los diferentes mercados en los que queremos, llegando con más facilidad al usuario final. El hecho de poder mantener unos costes operacionales relativamente bajos estando a la vez presentes en todo el mundo posibilita poder ofrecer precios competitivos. Si, además, es la propia marca la que está suministrando los productos, siempre genera más confianza la compra directa al fabricante, relegando a un segundo plano un posible incremento del coste de la compra.

Índice de Contenido

| | |
|--|----|
| Índice de Contenido | 4 |
| Índice de Tablas..... | 10 |
| Índice de Ilustraciones | 15 |
| 1. Introducción y objetivos..... | 17 |
| 1.1 Introducción | 17 |
| 1.2 Alcance del proyecto (Objetivos) | 17 |
| 2. Estado del Arte | 19 |
| 2.1 Plataforma como servicio (PAAS)..... | 19 |
| 2.1.1 Shopify..... | 20 |
| 2.1.2 Bigcommerce..... | 20 |
| 2.1.3 Magento Go (extinto)..... | 20 |
| 2.2 Plataformas auto gestionadas y alojadas..... | 21 |
| 2.2.1 Woocommerce..... | 21 |
| 2.2.2 Prestashop..... | 22 |
| 2.2.3 Magento | 23 |
| 2.3 Conclusiones..... | 24 |
| 3. Documentación | 28 |
| 3.1 Requisitos | 28 |
| 3.1.1 Requisitos de usuario | 29 |
| 3.1.2 Requisitos de software..... | 33 |
| 3.1.2.1 Requisitos del sistema..... | 33 |
| 3.1.2.2 Requisitos de funcionalidad del sistema..... | 35 |
| 3.1.2.3 Requisitos de interfaz del sistema | 44 |
| 3.1.3 Requisitos de restricción | 45 |
| 3.1.4 Requisitos de diseño del sistema hardware..... | 46 |
| 3.2 Desarrollo de software | 48 |
| 3.2.1 Diagrama de casos de uso..... | 48 |
| 3.2.1.1 Nivel 0..... | 48 |
| 3.2.1.2 Nivel 1..... | 49 |
| 3.2.2 Casos de Uso | 50 |
| 3.2.3 Diagramas de Secuencia..... | 57 |
| 3.2.3.1 Diagrama de Secuencia – Caso de Uso 005 - Registro | 57 |

| | |
|---|----|
| 3.2.3.2 Diagrama de Secuencia – Caso de Uso 001 - Compra..... | 58 |
| 3.2.4 Diagrama de clases..... | 60 |
| 3.3. Infraestructura de Servidor | 61 |
| 3.3.1 Sistema Propietario o Sistema Externalizado..... | 61 |
| 3.3.1.1 Sistema Propietario | 61 |
| 3.3.1.2 Sistema Externalizado | 61 |
| 3.3.2 Servidor VPS | 63 |
| 3.3.2.1 Tradicional vs Cloud | 63 |
| 3.3.2.2 Escalado VPS..... | 63 |
| 3.3.2.2.1 Escalado vertical..... | 64 |
| 3.3.2.2.2 Escalado horizontal | 64 |
| 3.3.2.2.2.1 Escalado horizontal en servidor web | 65 |
| 3.3.2.2.2.1.1 Control de sesiones | 65 |
| 3.3.2.2.2.1.2 Acceso caché | 66 |
| 3.3.2.2.2.1.3 Acceso a imágenes | 66 |
| 3.3.2.2.2.1.4 Deployment / despliegue | 66 |
| 3.3.2.2.2.2 Escalado horizontal en base de datos..... | 67 |
| 3.3.2.2.2.2.1 Escalado horizontal en base de datos: Master/Slave | 67 |
| 3.3.2.2.2.2.2 Escalado horizontal en base de datos: Master/Master | 68 |
| 3.3.2.2.2.2.3 Escalado horizontal en base de datos: Particionado funcional..... | 69 |
| 3.3.2.2.2.2.4 Escalado horizontal en base de datos: database sharding | 69 |
| 3.3.3 Software | 71 |
| 3.3.3.1 Sistema Operativo | 71 |
| 3.3.3.2 Servidor Web..... | 71 |
| 3.3.3.3 PHP | 72 |
| 3.3.3.4 Sistema Gestor de Bases de Datos (SGBD)..... | 72 |
| 3.3.3.4.1 MySQL | 73 |
| 3.4 Proveedor de Servicios..... | 74 |
| 3.4.1 Amazon Web Services | 74 |
| 3.4.1.1 Escalado vertical..... | 77 |
| 3.4.1.1.1 Esquema básico..... | 77 |
| 3.4.1.1.2 Esquema básico en VPC | 78 |
| 3.4.1.1.3 Esquema servidor Web y Base de datos en diferentes nodos..... | 79 |
| 3.4.1.2 Escalado horizontal | 80 |

| | |
|--|-----|
| 3.4.1.2.1 Esquema escalado horizontal Servidor Web y Servidor de Base de datos .. | 80 |
| 3.4.1.2.2 Esquema con Autoescalado y RDS | 81 |
| 3.4.1.3 Base de datos EC2 vs RDS..... | 83 |
| 3.4.1.4 Diseño final..... | 84 |
| 3.5 Metodología de desarrollo..... | 85 |
| 3.5.1 Herramientas de desarrollo | 85 |
| 3.5.1.1 Vagrant y Puppuppet..... | 86 |
| 3.5.2 Entornos de desarrollo | 86 |
| 3.5.2.1 Desarrollo | 86 |
| 3.5.2.2 Preproducción | 86 |
| 3.5.2.3 Producción..... | 87 |
| 3.6 LOPD | 88 |
| 3.6.1 LOPD en nuestro proyecto | 88 |
| 4. Implementación / Resultado..... | 90 |
| 4.1 Requisitos del proyecto..... | 90 |
| 4.2 Implementación de la arquitectura diseñada | 92 |
| 4.2.1 Región..... | 92 |
| 4.2.2 Zonas de disponibilidad..... | 92 |
| 4.2.3 Entorno VPC | 92 |
| 4.2.3.1 Creación de la VPC | 93 |
| 4.2.4 Instancia RDS (Base de datos) | 94 |
| 4.2.5 EC2..... | 96 |
| 4.2.5.1 Instancias servidor Web | 97 |
| 4.2.5.1.1 Instalación PHP..... | 98 |
| 4.2.5.1.2 Instalación Apache | 98 |
| 4.2.5.1.3 Espacio de intercambio o Swap | 100 |
| 4.2.5.2 Instancia NFS | 101 |
| 4.2.5.2.1 Single Point Of Failure en NFS | 101 |
| 4.2.5.2.1.1 Autoescalado NFS..... | 101 |
| 4.2.5.2.1.2 GlusterFS (Gluster File System) | 102 |
| 4.2.5.2.1.3 S3FS (S3 File System) | 102 |
| 4.2.5.2.1.4 Redis..... | 103 |
| 4.2.5.2.1.5 Elección final: ElasticCache (Redis) + S3FS | 103 |
| 4.2.6 Modificación del sistema. Sustitución NFS por instancia ElasticCache Redis + S3FS104 | |

| | |
|---|-----|
| 4.2.7 ELB (Elastic Load Balancer) | 106 |
| 4.2.7.1 Healthy Check..... | 107 |
| 4.2.7.2 Instalación Certificado SSL | 107 |
| 4.2.7.3 ELB en instancias Web..... | 108 |
| 4.2.8 CloudWatch | 109 |
| 4.2.8.1 Alarmas..... | 109 |
| 4.2.9 Autoscalling | 110 |
| 4.2.9.1 Autoescalado Web | 110 |
| 4.2.9.1.1 Creación de configuración de inicio | 110 |
| 4.2.9.1.2 Creación del grupo de autoescalado..... | 111 |
| 4.2.9.1.2.1 Configuración básica | 111 |
| 4.2.9.1.2.2 Configuración políticas de escalado | 112 |
| 4.2.10 Grupos de seguridad | 112 |
| 4.2.10.1. Grupo de Seguridad para las instancias RDS..... | 113 |
| 4.2.10.2 Grupo de seguridad para el balanceador web..... | 113 |
| 4.2.10.3 Grupo de seguridad para las instancias Web..... | 113 |
| 4.2.10.4 Grupo de seguridad para las instancias Redis | 114 |
| 4.2.11 Route53 | 114 |
| 4.2.12 CloudFront..... | 115 |
| 4.3 Desarrollo Magento | 116 |
| 4.3.1 Instalación | 116 |
| 4.3.1.1 Control de versiones | 117 |
| 4.3.2 Configuración e implantación | 117 |
| 4.3.2.1 Configuración con balanceador web..... | 117 |
| 4.3.2.1.1 Dirección IP origen | 117 |
| 4.3.2.1.2 Conexión segura (SSL) | 118 |
| 4.3.2.2 Geolocalización del cliente..... | 119 |
| 4.3.2.3 Configuración interna de Magento | 121 |
| 4.3.2.3.1 Monedas..... | 121 |
| 4.3.2.3.2 Idioma..... | 122 |
| 4.3.2.3.3 Impuestos..... | 122 |
| 4.3.2.3.4 Métodos de pago | 122 |
| 4.3.2.3.5 Sistema de envío | 123 |
| 4.3.2.3.6 Control de Stock | 123 |

| | |
|---|-----|
| 4.3.2.3.7 Cron | 124 |
| 4.3.2.3.8 Configuración de envío de emails | 124 |
| 4.3.2.3.10 Sesiones de usuario..... | 125 |
| 4.3.2.3.11 Optimización de Magento..... | 126 |
| 4.3.2.3.11.1 Caché | 126 |
| 4.3.2.3.11.2 Flat Catalog Product / Flat Catalog Category | 128 |
| 4.3.2.3.11.3 CDN..... | 129 |
| 4.3.2.5 Deployment..... | 131 |
| 4.3.2.6 Configuración final | 132 |
| 4.3.2.6.1 Gzip..... | 132 |
| 4.3.2.6.2 Expire Headers | 132 |
| 4.3.4 Pruebas y validación..... | 133 |
| 4.3.4.1 Pruebas..... | 133 |
| 4.3.4.1.1 Prueba 1 | 133 |
| 4.3.4.1.2 Prueba 2 | 140 |
| 4.3.4.1.3 Prueba 3 | 146 |
| 4.3.4.1.4 Prueba Autoescalado | 151 |
| 4.3.4.1.4.1 Configuración | 151 |
| 4.3.4.1.4.2 Estado inicial..... | 152 |
| 4.3.4.1.4.3 Iniciando test de estrés | 153 |
| 4.3.4.1.4.4 Incrementando recursos | 154 |
| 4.3.4.1.4.4.1 Primer incremento | 154 |
| 4.3.4.1.4.4.2 Segundo incremento | 156 |
| 4.3.4.1.4.5 Decrementando recursos..... | 157 |
| 4.3.4.1.5 Prueba acceso desde diferentes países | 159 |
| 4.3.4.1.5.1 Website Europa | 161 |
| 4.3.4.1.5.2 Website America | 162 |
| 4.3.4.1.5.3 Website Asia..... | 163 |
| 4.3.4.1.6 Conclusiones | 164 |
| 4.3.4.2 Validación de los requisitos de usuario..... | 165 |
| 4.4 Problemas encontrados | 173 |
| 5. Gestión del proyecto | 174 |
| 5.1 Planificación inicial | 174 |
| 5.2 Planificación final | 175 |

| | |
|--|-----|
| 5.3 Presupuesto | 176 |
| 5.3.1 Presupuesto Desarrollo | 176 |
| 5.3.2 Presupuesto tecnología AWS | 177 |
| 5.3.3 Presupuesto Final | 178 |
| 6. Conclusiones y trabajos futuros | 179 |
| 6.1 Conclusiones..... | 179 |
| 6.2 Trabajos futuros | 179 |
| 7. Bibliografía y Referencias | 180 |
| 7.1 Bibliografía | 180 |
| 7.2 Referencias..... | 180 |
| 8. Licencias | 200 |

Índice de Tablas

| | |
|--|----|
| Tabla 1: Shopify [7]..... | 20 |
| Tabla 2: Bigcommerce [8]..... | 20 |
| Tabla 3: Magento Go [9] | 20 |
| Tabla 4: Woocommerce [10]..... | 21 |
| Tabla 5: Prestashop [11]..... | 22 |
| Tabla 6: Magento [12] | 23 |
| Tabla 7: Plantilla requisitos | 28 |
| Tabla 8: Requisito Usuario 001..... | 29 |
| Tabla 9: Requisito Usuario 002..... | 29 |
| Tabla 10: Requisito Usuario 003 | 29 |
| Tabla 11: Requisito Usuario 004 | 30 |
| Tabla 12: Requisito Usuario 005 | 30 |
| Tabla 13: Requisito Usuario 006 | 30 |
| Tabla 14: Requisito Usuario 007 | 30 |
| Tabla 15: Requisito Usuario 008 | 30 |
| Tabla 16: Requisito Usuario 009 | 31 |
| Tabla 17: Requisito Usuario 010 | 31 |
| Tabla 18: Requisito Usuario 011 | 31 |
| Tabla 19: Requisito Usuario 012 | 31 |
| Tabla 20: Requisito Usuario 013 | 31 |
| Tabla 21: Requisito Usuario 014 | 32 |
| Tabla 22: Requisito Usuario 015 | 32 |
| Tabla 23: Requisito Usuario 016 | 32 |
| Tabla 24: Requisito Usuario 017 | 32 |
| Tabla 25: Requisito Usuario 018 | 32 |
| Tabla 26: Requisito Usuario 019 | 33 |
| Tabla 27: Requisito Usuario 020 | 33 |
| Tabla 28: Requisito Usuario 021 | 33 |
| Tabla 29: Requisito Software de Sistema 001..... | 33 |
| Tabla 30: Requisito Software de Sistema 002..... | 34 |
| Tabla 31: Requisito Software de Sistema 003..... | 34 |
| Tabla 32: Requisito Software de Sistema 004..... | 34 |

| | |
|---|----|
| Tabla 33: Requisito Software de Sistema 005..... | 34 |
| Tabla 34: Requisito Software de Sistema 006..... | 35 |
| Tabla 35: Requisito Software de Sistema 007..... | 35 |
| Tabla 36: Requisito Software de Funcionalidad del Sistema 001 | 35 |
| Tabla 37: Requisito Software de Funcionalidad del Sistema 002 | 35 |
| Tabla 38: Requisito Software de Funcionalidad del Sistema 003 | 36 |
| Tabla 39: Requisito Software de Funcionalidad del Sistema 004 | 36 |
| Tabla 40: Requisito Software de Funcionalidad del Sistema 005 | 36 |
| Tabla 41: Requisito Software de Funcionalidad del Sistema 006 | 36 |
| Tabla 42: Requisito Software de Funcionalidad del Sistema 007 | 36 |
| Tabla 43: Requisito Software de Funcionalidad del Sistema 008 | 37 |
| Tabla 44: Requisito Software de Funcionalidad del Sistema 009 | 37 |
| Tabla 45: Requisito Software de Funcionalidad del Sistema 010 | 37 |
| Tabla 46: Requisito Software de Funcionalidad del Sistema 011 | 37 |
| Tabla 47: Requisito Software de Funcionalidad del Sistema 012 | 37 |
| Tabla 48: Requisito Software de Funcionalidad del Sistema 013 | 38 |
| Tabla 49: Requisito Software de Funcionalidad del Sistema 014 | 38 |
| Tabla 50: Requisito Software de Funcionalidad del Sistema 015 | 38 |
| Tabla 51: Requisito Software de Funcionalidad del Sistema 016 | 38 |
| Tabla 52: Requisito Software de Funcionalidad del Sistema 017 | 38 |
| Tabla 53: Requisito Software de Funcionalidad del Sistema 018 | 39 |
| Tabla 54: Requisito Software de Funcionalidad del Sistema 019 | 39 |
| Tabla 55: Requisito Software de Funcionalidad del Sistema 020 | 39 |
| Tabla 56: Requisito Software de Funcionalidad del Sistema 021 | 39 |
| Tabla 57: Requisito Software de Funcionalidad del Sistema 022 | 39 |
| Tabla 58: Requisito Software de Funcionalidad del Sistema 023 | 40 |
| Tabla 59: Requisito Software de Funcionalidad del Sistema 024 | 40 |
| Tabla 60: Requisito Software de Funcionalidad del Sistema 025 | 40 |
| Tabla 61: Requisito Software de Funcionalidad del Sistema 026 | 40 |
| Tabla 62: Requisito Software de Funcionalidad del Sistema 027 | 41 |
| Tabla 63: Requisito Software de Funcionalidad del Sistema 028 | 41 |
| Tabla 64: Requisito Software de Funcionalidad del Sistema 029 | 41 |
| Tabla 65: Requisito Software de Funcionalidad del Sistema 030 | 41 |
| Tabla 66: Requisito Software de Funcionalidad del Sistema 031 | 41 |

| | |
|---|----|
| Tabla 67: Requisito Software de Funcionalidad del Sistema 032 | 42 |
| Tabla 68: Requisito Software de Funcionalidad del Sistema 033 | 42 |
| Tabla 69: Requisito Software de Funcionalidad del Sistema 034 | 42 |
| Tabla 70: Requisito Software de Funcionalidad del Sistema 035 | 42 |
| Tabla 71: Requisito Software de Funcionalidad del Sistema 036 | 43 |
| Tabla 72: Requisito Software de Funcionalidad del Sistema 037 | 43 |
| Tabla 73: Requisito Software de Funcionalidad del Sistema 038 | 43 |
| Tabla 74: Requisito Software de Funcionalidad del Sistema 039 | 43 |
| Tabla 75: Requisito Software de Funcionalidad del Sistema 040 | 43 |
| Tabla 76: Requisito Software de Funcionalidad del Sistema 041 | 44 |
| Tabla 77: Requisito Software de Funcionalidad del Sistema 042 | 44 |
| Tabla 78: Requisito Software de Funcionalidad del Sistema 043 | 44 |
| Tabla 79: Requisito Software de Funcionalidad del Sistema 044 | 44 |
| Tabla 80: Requisito Software de Interfaz del Sistema 001 | 45 |
| Tabla 81: Requisito Software de Interfaz del Sistema 002 | 45 |
| Tabla 82: Requisito de Restricción 001 | 45 |
| Tabla 83: Requisito de Restricción 002 | 45 |
| Tabla 84: Requisito de Restricción 003 | 46 |
| Tabla 85: Requisito de Diseño del Sistema Hardware 001..... | 46 |
| Tabla 86: Requisito de Diseño del Sistema Hardware 002..... | 46 |
| Tabla 87: Requisito de Diseño del Sistema Hardware 003..... | 47 |
| Tabla 88: Requisito de Diseño del Sistema Hardware 004..... | 47 |
| Tabla 89: Requisito de Diseño del Sistema Hardware 005..... | 47 |
| Tabla 90: Cuadro ejemplo casos de uso..... | 50 |
| Tabla 91: Caso de uso - 001..... | 51 |
| Tabla 92: Caso de uso - 002..... | 51 |
| Tabla 93: Caso de uso - 003..... | 52 |
| Tabla 94: Caso de uso - 004..... | 52 |
| Tabla 95: Caso de uso - 005..... | 52 |
| Tabla 96: Caso de uso - 006..... | 53 |
| Tabla 97: Caso de uso - 007..... | 53 |
| Tabla 98: Caso de uso - 008..... | 53 |
| Tabla 99: Caso de uso – 009 | 54 |
| Tabla 100: Caso de uso - 010..... | 54 |

| | |
|--|-----|
| Tabla 101: Caso de uso - 011..... | 55 |
| Tabla 102: Caso de uso – 012..... | 55 |
| Tabla 103: Caso de uso - 013..... | 55 |
| Tabla 104: Caso de uso - 014..... | 56 |
| Tabla 105: Caso de uso - 015..... | 56 |
| Tabla 106: Caso de uso - 016..... | 56 |
| Tabla 107: Herramientas de desarrollo..... | 85 |
| Tabla 108: Ventajas – Desventajas Magento AWS | 91 |
| Tabla 109: Parámetros RDS..... | 96 |
| Tabla 110: Tabla resumen características manejadores PHP | 99 |
| Tabla 111: Plantilla validación requisitos..... | 165 |
| Tabla 112: Validación de requisito de usuario 001 | 166 |
| Tabla 113: Validación de requisito de usuario 002 | 166 |
| Tabla 114: Validación de requisito de usuario 003 | 166 |
| Tabla 115: Validación de requisito de usuario 004 | 166 |
| Tabla 116: Validación de requisito de usuario 005 | 167 |
| Tabla 117: Validación de requisito de usuario 006 | 167 |
| Tabla 118: Validación de requisito de usuario 007 | 168 |
| Tabla 119: Validación de requisito de usuario 008 | 168 |
| Tabla 120: Validación de requisito de usuario 009 | 169 |
| Tabla 121: Validación de requisito de usuario 010 | 169 |
| Tabla 122: Validación de requisito de usuario 011 | 169 |
| Tabla 123: Validación de requisito de usuario 012 | 170 |
| Tabla 124: Validación de requisito de usuario 013 | 170 |
| Tabla 125: Validación de requisito de usuario 014 | 170 |
| Tabla 126: Validación de requisito de usuario 015 | 170 |
| Tabla 127: Validación de requisito de usuario 016 | 171 |
| Tabla 128: Validación de requisito de usuario 017 | 171 |
| Tabla 129: Validación de requisito de usuario 018 | 171 |
| Tabla 130: Validación de requisito de usuario 019 | 172 |
| Tabla 131: Validación de requisito de usuario 020 | 172 |
| Tabla 132: Validación de requisito de usuario 021 | 172 |
| Tabla 133: Presupuesto desarrollo | 176 |
| Tabla 134: Coste Recursos Humanos | 177 |

| | |
|--|-----|
| Tabla 135: Presupuesto sistema AWS..... | 178 |
| Tabla 136: Presupuesto total | 178 |
| Tabla 137: Tabla resumen licencias | 200 |

Índice de Ilustraciones

| | |
|--|-----|
| Ilustración 1: Comparación Magento y Prestashop [14]..... | 25 |
| Ilustración 2: Top 10k y 100k sitios web [15] | 26 |
| Ilustración 3: Top 1 millón sitios web y todo internet | 27 |
| Ilustración 4: Diagrama de casos de uso – Nivel 0 | 48 |
| Ilustración 5: Diagrama de casos de uso – Nivel 1 | 49 |
| Ilustración 6: Diagrama de Secuencia - Registro | 57 |
| Ilustración 7: Diagrama de secuencia – Compra – Parte1 | 58 |
| Ilustración 8 Diagrama de secuencia – Compra – Parte2 | 59 |
| Ilustración 9: Diagrama de Clases | 60 |
| Ilustración 10: Diagrama resumen “Infraestructura de Servidor” | 62 |
| Ilustración 11: Diagrama resumen "Escalado en servidor VPS" | 70 |
| Ilustración 12: Diagrama simple Servidor Web y BD..... | 77 |
| Ilustración 13: Diagrama Servidor Web y BD simple en VPC | 78 |
| Ilustración 14: Diagrama Servidor Web y BD en diferentes nodos..... | 79 |
| Ilustración 15: Diagrama escalado horizontal Servidor Web y Servidor de Base de datos | 80 |
| Ilustración 16: Diagrama Sistema Autoescable y tolerante a fallos en AWS | 82 |
| Ilustración 17: Diagrama VPC..... | 94 |
| Ilustración 18: Diagrama modificado Sistema Autoescable y tolerante a fallos en AWS | 105 |
| Ilustración 19: Política de seguridad entrante para el balanceador web | 113 |
| Ilustración 20: Funcionamiento caché dos niveles Magento..... | 127 |
| Ilustración 21: Model EAV en Magento [149]..... | 128 |
| Ilustración 22: Configuración CDN Magento para conexiones HTTP | 129 |
| Ilustración 23: Configuración CDN Magento para conexiones HTTPS | 130 |
| Ilustración 24: CDN en herramientas para desarrollador - 1 | 130 |
| Ilustración 25: CDN en herramientas para desarrollador - 2 | 131 |
| Ilustración 26: Log Test sin ninguna caché activada | 134 |
| Ilustración 27: Aumento de errores 500 ELB por tiempo de sesión expirado. Prueba 1 | 134 |
| Ilustración 28: Carga servidor Htop. Prueba 1 | 135 |
| Ilustración 29: Gráfico de CloudWatch del porcentaje de carga de las CPU de los dos servidores web. Prueba 1 | 135 |
| Ilustración 30: Métrica CloudWatch del servidor Redis. Prueba 1 | 136 |
| Ilustración 31: Métricas RDS (base de datos). Prueba 1 | 137 |
| Ilustración 32: Test rendimiento Gtmetrix.com. Prueba 1 | 138 |
| Ilustración 33: Test rendimiento WebPageTest.com. Prueba 1 | 138 |
| Ilustración 34: Waterfall WebPageTest.com. Prueba 1 | 139 |
| Ilustración 35: Log Test con la caché de Magento activada. Prueba 2 | 140 |
| Ilustración 36: Carga servidor Htop. Prueba 2 | 140 |
| Ilustración 37: Gráfico de CloudWatch del porcentaje de carga de las CPU de los servidores web. Prueba 2 | 141 |
| Ilustración 38: Métrica CloudWatch del servidor Redis. Prueba 2 | 142 |
| Ilustración 39: Métricas RDS (base de datos). Prueba 2 | 143 |
| Ilustración 40: Gráfica ELB sin más errores 500 por saturación. Prueba 2. | 144 |

| | |
|--|-----|
| Ilustración 41: Test rendimiento Gtmetrix.com. Prueba 2 | 144 |
| Ilustración 42: Test rendimiento WebPageTest.com. Prueba 2 | 145 |
| Ilustración 43: Log Test con la caché de Magento activada. Prueba 3 | 146 |
| Ilustración 44: Carga servidor Htop. Prueba 3 | 146 |
| Ilustración 45: Gráfico de CloudWatch del porcentaje de carga de las CPU de los servidores web. Prueba 3 | 147 |
| Ilustración 46: Métrica CloudWatch del servidor Redis. Prueba 3 | 147 |
| Ilustración 47: Métricas RDS (base de datos). Prueba 3 | 148 |
| Ilustración 48: Test rendimiento Gtmetrix.com. Prueba 3 | 149 |
| Ilustración 49: Test rendimiento WebPageTest.com. Prueba 3 | 149 |
| Ilustración 50: Waterfall WebPageTest.com. Prueba 3 | 150 |
| Ilustración 51: Situación inicial Auto-escalado..... | 152 |
| Ilustración 52: Situación inicial Auto-escalado. Alarmas OK..... | 152 |
| Ilustración 53: Situación inicial Autoescalado. Elastic Load Balancer..... | 152 |
| Ilustración 54: Auto-escalado.Carga servidor Htop..... | 153 |
| Ilustración 55: Auto-escalado. Alarmas OK..... | 153 |
| Ilustración 56: Auto-escalado. Alarma incremento de recursos disparada..... | 154 |
| Ilustración 57: Auto-escalado. ELB con instancia pendiente de healthy check..... | 155 |
| Ilustración 58: Auto-escalado. Instancia a la espera de ser activada por ELB..... | 155 |
| Ilustración 59: Auto-escalado. Todas las instancias recibiendo tráfico..... | 156 |
| Ilustración 60: Auto-escalado. Panel auto-escalado con tres instancias..... | 156 |
| Ilustración 61: Auto-escalado. Carga CPU cuatro instancias..... | 156 |
| Ilustración 62: Auto-escalado. ELB con 4 instancias..... | 157 |
| Ilustración 63: Auto-escalado. Panel EC2 con 4 instancias..... | 157 |
| Ilustración 64: Auto-escalado. Baja carga CPU cuatro instancias..... | 157 |
| Ilustración 65: Auto-escalado. Alarma decremento de recursos disparada..... | 158 |
| Ilustración 66: Auto-escalado. Panel EC2 con eliminando instancia-1..... | 158 |
| Ilustración 67: Auto-escalado. Panel EC2 con eliminando instancia-2..... | 159 |
| Ilustración 68: Auto-escalado. Panel EC2 con eliminando instancia-3..... | 159 |
| Ilustración 69: Home Tienda Europa Island | 161 |
| Ilustración 70: Home Tienda America Island | 162 |
| Ilustración 71: Home Tienda Asia Island | 163 |
| Ilustración 72: GANTT - Planificación inicial..... | 174 |
| Ilustración 73: GANTT - Planificación inicial - camino crítico | 174 |
| Ilustración 74: GANTT - Desviación planificación inicial | 175 |
| Ilustración 75: Desviación planificación inicial - camino crítico | 175 |

1. Introducción y objetivos

1.1 Introducción

Desde nuestros comienzos, el comercio ha supuesto uno de los pilares fundamentales de la evolución de las distintas civilizaciones, marcando diferencias aquellas que basaban su economía en el comercio a gran escala con distintas naciones.

En la época actual, ya no son necesarias grandes travesías en barco o viajes a través de distintos continentes para hacer llegar nuestros productos a cualquier parte del mundo. Cualquier persona puede adquirir el producto que desea desde el salón de su casa.

El siguiente paso está siendo conseguir que cualquier persona no pueda ya sólo comprar en cualquier parte del mundo si no, con una limitada inversión, tener la posibilidad de llegar a vender sus productos en cualquier rincón del mundo.

Gracias al uso de software libre, se puede conseguir agrupar en un único sistema todos los procesos relacionados con la venta por internet a nivel global, conteniendo la inversión necesaria.

1.2 Alcance del proyecto (Objetivos)

El proyecto surge con la premisa de poder ser asumido por una pequeña empresa cuyo objetivo sea la venta online de productos. Por lo tanto, el proyecto debe cubrir al menos los siguientes requisitos:

- Venta a nivel mundial: El sistema tiene que tener la capacidad de poder gestionar la venta a nivel mundial de los productos. Esto implica:
 - Varios idiomas.
 - Varias monedas (informativo).
 - Distintas formas de pago
 - Transporte
 - Tiendas diferentes por país. Cada tienda puede tener diferente diseño, productos, o incluso, diferentes precios en los mismos productos.
- La configuración del sistema, su gestión, mantenimiento y evolución de cara al futuro deben de ser lo más sencillas posible, por lo que se requiere un Sistema Gestor de Contenidos[1] (CMS) de código abierto (Open Source[2]), con una amplia implantación en el mercado y una gran comunidad. Al hacer uso de un Sistema Gestor de Contenidos

los costes de implementación del sistema no son altos a la vez que se asegura una facilidad de uso de cara al cliente. Al ser de código abierto se permite un desarrollo y personalización acorde con las necesidades actuales o futuras del cliente. El tener una amplia implantación en el mercado y una gran comunidad lo dota de una gran cantidad de aplicaciones o módulos que permitan ampliar las funcionalidades del sistema, una buena documentación, un buen soporte antes vulnerabilidades y un desarrollo de futuro.

- En la medida de lo posible lo ideal sería tener un sistema “todo en uno”, que permitiera desde un solo panel, poder controlar:
 - Gestión de todas las tiendas
 - Ventas
 - Facturación
 - Estadísticas
 - Gestión de clientes.

Se busca una solución que evite o reduzca la necesidad de compra de diferentes sistemas para la gestión de todos los elementos. De esta forma, reducimos la curva de aprendizaje que se necesita para el control del mismo y reducimos los costes.

2. Estado del Arte

Actualmente existen numerosas soluciones de software destinadas al comercio electrónico. Hay dos tipos claramente diferenciados. Las plataformas de software propietario que se comercializan como SAAS[3] (Software as a Service) y las de software de código abierto. Las primeras no cumplen directamente con el requisito de que la plataforma sea Open Source, pero creemos que es importante hablar sobre ellas y mencionar algunos de sus puntos fuertes y de sus puntos débiles.

2.1 Plataforma como servicio (PAAS)

El sistema PASS[4] (Platform as a Service) tiene un modelo de negocio orientado al pago de una cuota mensual y un porcentaje por cada compra realizada.

Este tipo de sistemas se ocupa de la gestión de la plataforma de comercio electrónico en sí, siendo el usuario únicamente responsable de la gestión de los productos y, en la medida en que se permita, de la personalización visual de la tienda.

Además, al estar diseñadas para un perfil de cliente lo más universal posible, suelen tener un diseño amigable y una configuración más sencilla.

Entre sus mayores desventajas está la poca personalización. Si bien es cierto que este tipo de plataformas suelen tener su propia “tienda” de Apps las cuales añaden funcionalidades (redes sociales, SEO[5]...) siempre nos encontramos ante la limitación de que lo que exista lo que necesitamos.

A la hora de modificar el diseño, según la plataforma tendremos más o menos opciones, pero siempre estaremos muy limitados.

En este tipo de soluciones, el software no es plenamente flexible a tus necesidades si no que hemos de adaptarnos al él.

A continuación, presentamos algunos ejemplos con sus ventajas y desventajas [6].

2.1.1 Shopify

| Ventajas | Desventajas |
|--|---|
| <ul style="list-style-type: none">✓ Fácilmente configurable✓ Panel de gestión sencillo e intuitivo✓ Sólo hay que gestionar los productos y las compras, no la plataforma✓ Variedad de módulos (gratuitos y de pago) | <ul style="list-style-type: none">✗ Personalizaciones muy limitadas✗ Coste mensual y porcentaje por compra✗ Desarrollo muy limitado✗ No permite multi-tienda |

Tabla 1: Shopify [7]

2.1.2 Bigcommerce

| Ventajas | Desventajas |
|---|--|
| <ul style="list-style-type: none">✓ Fácilmente configurable✓ Panel de gestión sencillo e intuitivo✓ Solo hay que gestionar los productos y las compras, no la plataforma✓ Curva de aprendizaje pequeña | <ul style="list-style-type: none">✗ Menos flexible que Shopify✗ Coste mensual y porcentaje por compra✗ Desarrollo muy limitado✗ No permite multi-tienda |

Tabla 2: Bigcommerce [8]

2.1.3 Magento Go (extinto)

| Ventajas | Desventajas |
|--|--|
| <ul style="list-style-type: none">✓ La potencia de Magento sin las preocupaciones de la gestión del alojamiento.✓ Precios más competitivos que la competencia✓ Misma tienda de aplicaciones✓ Curva de aprendizaje pequeña | <ul style="list-style-type: none">✗ Menos flexible que Shopify✗ Coste mensual y porcentaje por compra✗ Desarrollo muy limitado✗ No permite multi-tienda |

Tabla 3: Magento Go [9]

2.2 Plataformas auto gestionadas y alojadas

En este tipo de soluciones la plataforma se suministra en formato de descarga y uno mismo ha de encargarse de la instalación en un entorno adecuado y de la gestión del mismo.

Existen cada vez más empresas que ofrecen soluciones de hosting especializadas o incluso con el propio software ya instalado. De esta forma, se tienen parte de las ventajas del sistema anterior y además se mantienen las opciones de las plataformas autogestionadas.

Las plataformas autogestionadas permiten un nivel de personalización muy superior. Esto no quiere decir que la empresa que necesite poner en marcha un comercio electrónico pueda acometer todo el proyecto por sí misma, pero sí que permite que empresas especializadas puedan ofrecer una tienda lo más a medida posible acorde a las necesidades y requisitos del cliente final sin llegar al desarrollo de un sistema completamente a medida.

Lo importante en estos casos siempre es elegir una plataforma con una gran comunidad de desarrolladores que permita, por un lado un conjunto de extensiones que añadan funcionalidades sin necesidad de requerir a un desarrollo propio (parecido a las Apps que hablábamos en el caso anterior) y por otro un soporte asegurado, documentación, desarrollo constante y solución de errores y vulnerabilidades.

Presentamos algunos ejemplos:

2.2.1 Woocommerce

| Ventajas | Desventajas |
|--|---|
| <ul style="list-style-type: none">✓ Fácil de instalar como plugin de Wordpress.✓ Manejo sencillo.✓ Permite personalización. Gran cantidad de plantillas✓ Código abierto | <ul style="list-style-type: none">✗ No deja de ser un plugin (Wordpress)✗ Poca potencia y escalabilidad.✗ No es recomendable como tienda en sí, si no como añadido a una web en la que se quiera vender algo. |

Tabla 4: Woocommerce [10]

2.2.2 Prestashop

| Ventajas | Desventajas |
|---|---|
| <ul style="list-style-type: none">✓ Bajo coste de implementación✓ Sencillez en instalación y uso.✓ Soporta múltiples tiendas.✓ Soporta múltiples idiomas.✓ Código abierto✓ BackOffice sencillo e intuitivo.✓ Rapidez de carga de las páginas.✓ Bajo consumo de recursos, por lo que no es imprescindible contar con un hosting dedicado.✓ Muy personalizable.✓ Gran cantidad de módulos (generalmente de pago) disponibles.✓ La Comunidad de Prestashop es muy activa y es muy usado en el mercado español. | <ul style="list-style-type: none">✗ Poca escalabilidad✗ La mayor parte de módulos y plantillas para personalizar nuestra tienda, son de pago.✗ No dispone de soporte técnico.✗ SEO onPage no tan potente✗ Actualizaciones muy frecuentes pero con dependencia técnica la hora de ser instaladas (sobre todo en una tienda en producción). |

Tabla 5: Prestashop [11]

2.2.3 Magento

| Ventajas | Desventajas |
|--|--|
| <ul style="list-style-type: none"> ✓ Muy personalizable. ✓ Un BackOffice muy potente y completo. ✓ Soporta múltiples tiendas ✓ Soporta múltiples monedas ✓ Soporta múltiples idiomas ✓ Tiene soporte técnico. ✓ Sistema robusto y seguro. ✓ SEO onPage potente y sencillo. ✓ Sistema de compra más sencillo y menos estricto. Gran posibilidad de modificación del mismo. ✓ Sistema muy robusto y muy seguro. ✓ Código abierto ✓ Gran opción de plantillas que permiten personalizar la tienda sin necesidad de grandes inversiones ni conocimientos. ✓ Muy buena comunidad de soporte, pero en inglés. Empieza a haberla en español ✓ Gran posibilidad de escalabilidad: <ul style="list-style-type: none"> ➤ A nivel de ventas. Tiene mayor capacidad que otros sistemas a la hora de manejar gran cantidad de productos y de transacciones. Permite albergar diferentes tiendas completamente diferentes, incluso con diferente catálogo. ➤ A nivel de internacionalización en diferentes mercados. Fácil gestión de diferentes idiomas y monedas. | <ul style="list-style-type: none"> ✗ Necesidad de recursos mayores ✗ Lenta curva de aprendizaje ✗ Mayor coste de implementación que otras plataformas, en parte debido a su complejidad pero también a su gran cantidad de opciones y escalabilidad ✗ Difícil de configurar debido a su gran poder de personalización y a su cantidad de opciones. Se le acusa de no tener una interfaz amigable |

Tabla 6: Magento [12]

2.3 Conclusiones

La primera decisión a tomar es el tipo de plataforma que usaremos.

Las soluciones PASS nos permiten crear tiendas sencillas en poco tiempo, sin necesidad de grandes conocimientos ni un equipo Sysadmin [7] que nos gestione su alojamiento, a cambio de una cuota mensual y un porcentaje por venta.

La problemática surge al no cumplir con varios de los requisitos del proyecto tales como que el sistema debía ser Open Source, personalizable tanto a nivel de diseño como de funcionalidades y multi-tienda, es decir, que a través de un solo panel poder controlar diferentes tiendas cuyos productos, diseño y hasta dominio puedan ser diferentes.

Por lo tanto, la decisión se ha de tomar entre las plataformas autogestionadas Magento, Prestashop y Woocommerce,

Woocommerce cumple con la mayor parte de los requisitos pero no deja de ser un plugin de Wordpress que no está pensado ni preparado para la configuración ni potencia necesaria para desarrollar este proyecto, si no para proyectos de pequeña envergadura, complejidad y sin grandes perspectivas de escalado.

La elección se ha de tomar entre Magento y Prestashop.

Prestashop es un sistema de código abierto potente, flexible, de fácil manejo y con una curva de aprendizaje reducida. Permite una alta personalización sin necesidad de ser un experto y el coste de creación de una tienda no es muy alto. Es potente y rápido y no consume grandes recursos, por lo que permite su instalación en entornos económicos.

Es ideal para proyectos de tamaño pequeño y medio, que no necesiten un desarrollo a medida grande.

Magento es una plataforma de código abierto muy potente, flexible y escalable. Es más compleja que Prestashop desde su misma instalación y su curva de aprendizaje es más lenta, tanto en lo que se refiere a su configuración y uso diario como en desarrollos a medida o de módulos que amplíen sus características. Es altamente recomendable el uso como mínimo de un servidor VPS.

Es más completo que el resto de sistemas y está diseñado para todo tipo de proyectos por lo que ahí radica su complejidad, pero lo dota de una capacidad de escalado que no tienen el resto de competidores.

No es casualidad que la mayor parte de las migraciones de plataforma se producen hacia Magento, ya que muchos proyectos son comenzados con sistemas más sencillos y económicos, pero que a la larga no cubren las necesidades de crecimiento.

Finalmente, Magento es el sistema elegido.

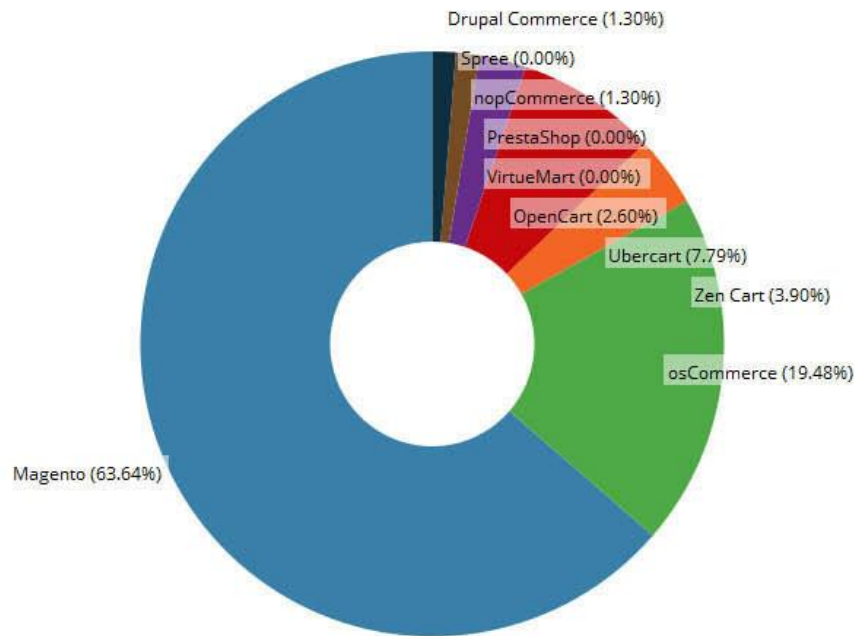
Algunos datos de interés:

Magento es usado por grandes empresas [13] como Samsung, Ford, Men's Health para sus tiendas de comercio electrónico lo cual nos permite ver sus posibilidades.

|  Magento™ Open Source eCommerce | VS |  PRESTASHOP The Best E-Commerce Experience |
|--|---------------------|---|
|  | EASY TO INSTALL |  |
|  | NUMBER OF FEATURES |  |
|  | SYSTEM REQUIREMENTS |  |
|  | COMPATIBILITY |  |
|  | EASY TO CUSTOMIZE |  |
|  | TECH SUPPORT |  |
|  | SEARCH FUNCTION |  |
|  | THEMES AVAILABLE |  |
|  | SEO OPTIONS |  |

Ilustración 1: Comparación Magento y Prestashop [14]

Top 10k Sites



Top 100k Sites

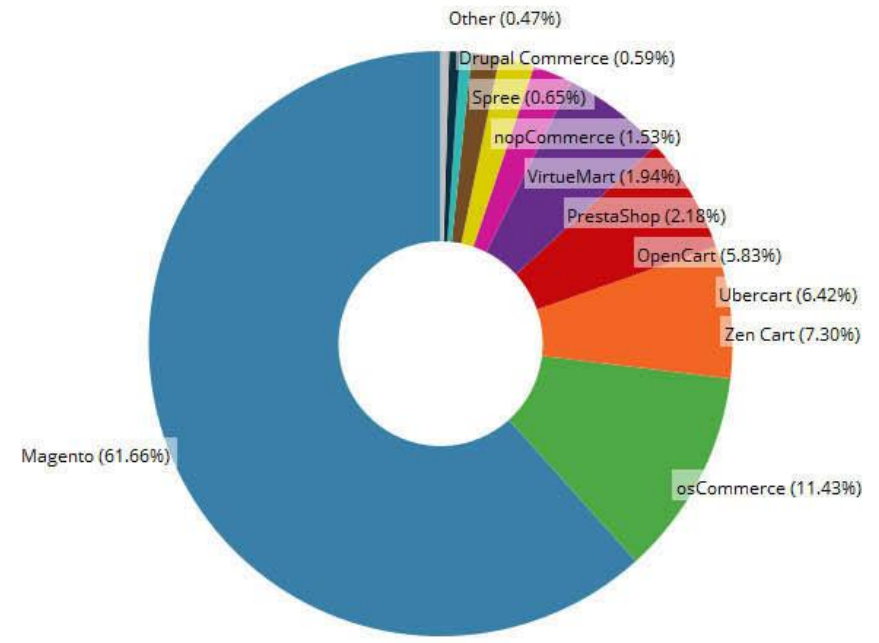
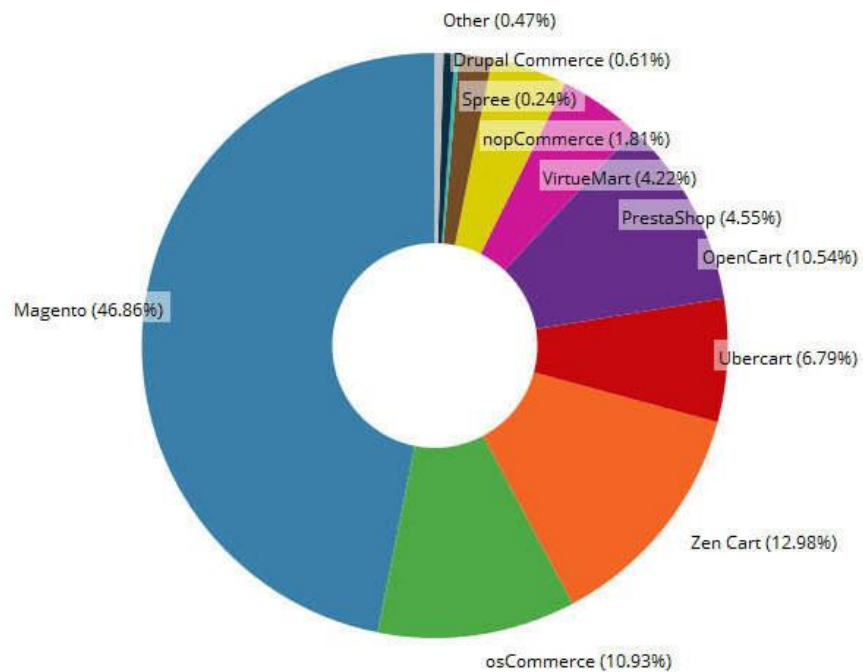


Ilustración 2: Top 10k y 100k sitios web [15]

Top Million Sites



The Entire Internet

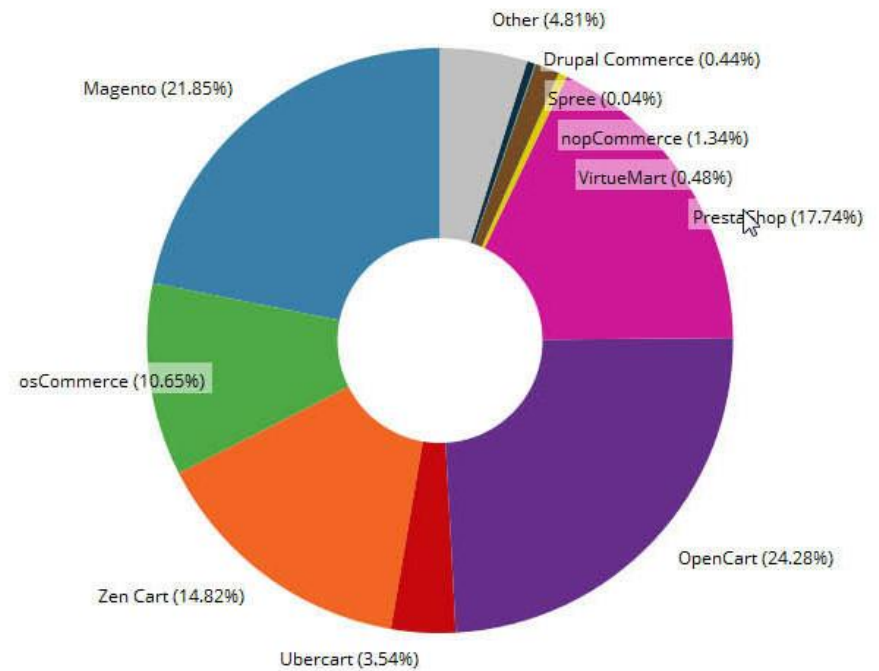


Ilustración 3: Top 1 millón sitios web y todo internet

3. Documentación

3.1 Requisitos

En este apartado especificaremos los requisitos obtenidos una vez se ha realizado el análisis del sistema y las necesidades del cliente. Los requisitos estarán divididos en Requisitos de usuario, Requisitos de restricción, Requisitos de software y Requisitos de diseño del sistema.

Para la definición de cada uno de los requisitos haremos uso de una “tabla plantilla” como la siguiente:

| Identificador de requisito | | | |
|----------------------------|--|-----------|--|
| Descripción | | | |
| Verificabilidad | | Prioridad | |
| Necesidad | | Fuente | |

Tabla 7: Plantilla requisitos

Descripción de los campos:

- Identificador de requisito: Campo único que identifica el requisito. Su nomenclatura varía en función del requisito.
 - Requisitos de usuario: RU-XXX. Las X hacen referencia al número de requisito.
 - Requisitos de restricción: RR-XXX
 - Requisitos de diseño del sistema hardware: RD-XXX
 - Requisitos de software: RS-X-YYY
 - Valores de X
 - ✓ S: para indicar un requisito del sistema.
 - ✓ F: para indicar un requisito de funcionalidad del sistema.
 - ✓ I: para indicar un requisito de interfaz del sistema.
- Descripción: Descripción del requisito
- Verificabilidad: Indica la capacidad de comprobación de que el requisito se ha introducido en el sistema. Valores posibles:
 - Alta: Comprobación clara y sin lugar a dudas.
 - Media: Comprobación posible, pero no de forma tan irrefutable.
 - Baja: Comprobación complicada o incluso imposible.
- Necesidad: representa el grado de necesidad de que ese requisito se incorpore al sistema. Valores posibles:
 - Esencial: Es imprescindible que el requisito sea introducido en el sistema.

- Deseable: No es obligatoria pero sí recomendable la introducción de dicho requisito.
- Opcional: La introducción del requisito es completamente opcional.
- **Prioridad:** Indica el grado de prioridad con el que el requisito debe ser implantado.
Valores posibles:
 - Alta: Requisito de prioridad alta.
 - Media: Requisito de prioridad alta.
 - Baja: Requisito de prioridad alta.
- **Fuente:** Representa el origen del requisito. Valores posibles:
 - Autor: El requisito lo ha establecido el autor del proyecto.
 - Cliente: El requisito lo ha establecido el cliente del proyecto.

3.1.1 Requisitos de usuario

| RU-001 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá elegir el idioma en el que visualizar la tienda entre varias opciones disponibles | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 8: Requisito Usuario 001

| RU-002 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá elegir la moneda en la que visualizar los precios de los productos entre varias opciones. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 9: Requisito Usuario 002

| RU-003 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario podrá registrarse en el sistema. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 10: Requisito Usuario 003

| RU-004 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario podrá iniciar sesión en el sistema | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 11: Requisito Usuario 004

| RU-005 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario registrado podrá visualizar su historial de pedidos. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 12: Requisito Usuario 005

| RU-006 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario podrá recuperar su carrito de la compra durante la duración de su cookie de sesión | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 13: Requisito Usuario 006

| RU-007 | | | |
|------------------------|------------------------------------|------------------|---------|
| Descripción | El usuario podrá comprar productos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 14: Requisito Usuario 007

| RU-008 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá acceder a la tienda desde su Smartphone con plena funcionalidad | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 15: Requisito Usuario 008

| RU-009 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá acceder a la tienda desde su tablet con plena funcionalidad | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 16: Requisito Usuario 009

| RU-010 | | | |
|------------------------|-------------------------------------|------------------|---------|
| Descripción | El usuario podrá comparar productos | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 17: Requisito Usuario 010

| RU-011 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá añadir productos a su lista de deseos | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 18: Requisito Usuario 011

| RU-012 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá realizar búsquedas de productos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 19: Requisito Usuario 012

| RU-013 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario tendrá acceso a la política de envíos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 20: Requisito Usuario 013

| RU-014 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario tendrá acceso a la política de privacidad | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 21: Requisito Usuario 014

| RU-015 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario tendrá acceso a la política de devoluciones | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 22: Requisito Usuario 015

| RU-016 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario tendrá acceso a la política de cookies | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 23: Requisito Usuario 016

| RU-017 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario se podrá suscribir a la newsletter de la tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 24: Requisito Usuario 017

| RU-018 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá navegar por las categorías de la tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 25: Requisito Usuario 018

| RU-019 | | | |
|------------------------|---|------------------|---------|
| Descripción | El usuario se podrá visualizar todos los productos de la tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 26: Requisito Usuario 019

| RU-020 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá comprar los productos desde la página de catálogo | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 27: Requisito Usuario 020

| RU-021 | | | |
|------------------------|--|------------------|---------|
| Descripción | El usuario podrá ver en todo momento un resumen del carro de la compra | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 28: Requisito Usuario 021

3.1.2 Requisitos de software

3.1.2.1 Requisitos del sistema

| RS-S-001 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema de comercio electrónico debe basarse en un Sistema Gestor de Contenidos (CMS) OpenSource | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 29: Requisito Software de Sistema 001

| RS-S-002 | | | |
|------------------------|---|------------------|---------|
| Descripción | El CMS no debe acarrar ningún coste por uso | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 30: Requisito Software de Sistema 002

| RS-S-003 | | | |
|------------------------|---|------------------|---------|
| Descripción | El CMS de comercio electrónico debe disponer de una documentación suficiente así como soporte por parte de la comunidad de usuarios | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 31: Requisito Software de Sistema 003

| RS-S-004 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema de comercio electrónico debe tener una amplia implantación a nivel mundial, lo que asegura su futuro | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 32: Requisito Software de Sistema 004

| RS-S-005 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema de comercio electrónico debe estar fortificado ante ataques de SQL Injection [16] y XSS [17] (Cross-site Scripting) | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 33: Requisito Software de Sistema 005

| RS-S-006 | | | |
|------------------------|--|------------------|------|
| Descripción | El sistema de comercio electrónico debe poder recibir actualizaciones y parches de seguridad que aseguren su futuro y su seguridad | | |
| Verificabilidad | Alta | Prioridad | Alta |

| | | | |
|------------------|----------|---------------|---------|
| Necesidad | Esencial | Fuente | Cliente |
|------------------|----------|---------------|---------|

Tabla 34: Requisito Software de Sistema 006

| RS-S-007 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder hacer uso de una conexión segura SSL/TLS [18] | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 35: Requisito Software de Sistema 007

3.1.2.2 Requisitos de funcionalidad del sistema

| RS-F-001 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe tener la capacidad de gestionar la venta de productos a nivel mundial | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 36: Requisito Software de Funcionalidad del Sistema 001

| RS-F-002 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar multi-tiendas | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 37: Requisito Software de Funcionalidad del Sistema 002

| RS-F-003 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe tener la capacidad de gestionar el acceso a las tiendas en función del país de origen del cliente | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 38: Requisito Software de Funcionalidad del Sistema 003

| RS-F-004 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe poder gestionar categorías de producto | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 39: Requisito Software de Funcionalidad del Sistema 004

| RS-F-005 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar distintas categorías de producto en cada tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 40: Requisito Software de Funcionalidad del Sistema 005

| RS-F-006 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar productos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 41: Requisito Software de Funcionalidad del Sistema 006

| RS-F-007 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar productos configurables, de forma que tengan diferentes opciones a elegir durante la compra | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 42: Requisito Software de Funcionalidad del Sistema 007

| RS-F-008 | | | |
|------------------------|--|------------------|------|
| Descripción | El sistema debe poder gestionar productos virtuales, esto es, productos que no requieran ningún tipo de envío o descarga | | |
| Verificabilidad | Alta | Prioridad | Alta |

| | | | |
|------------------|----------|---------------|---------|
| Necesidad | Esencial | Fuente | Cliente |
|------------------|----------|---------------|---------|

Tabla 43: Requisito Software de Funcionalidad del Sistema 008

| RS-F-009 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar la creación de packs de productos a partir de los productos ya existentes | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 44: Requisito Software de Funcionalidad del Sistema 009

| RS-F-010 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder permitir la asignación de diferentes productos por tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 45: Requisito Software de Funcionalidad del Sistema 010

| RS-F-011 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe tener un proceso de control de stock de los productos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 46: Requisito Software de Funcionalidad del Sistema 011

| RS-F-012 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir la activación o desactivación del control de stock por producto | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 47: Requisito Software de Funcionalidad del Sistema 012

| RS-F-013 | |
|--------------------|---|
| Descripción | El sistema debe permitir habilitar o deshabilitar un producto de forma individual |

| | | | |
|------------------------|----------|------------------|---------|
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 48: Requisito Software de Funcionalidad del Sistema 013

| RS-F-014 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir la asignación de un precio por producto | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 49: Requisito Software de Funcionalidad del Sistema 014

| RS-F-015 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir la asignación de un precio a cada producto en cada tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 50: Requisito Software de Funcionalidad del Sistema 015

| RS-F-016 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir resaltar productos como “nuevos” durante un periodo de tiempo | | |
| Verificabilidad | Alta | Prioridad | Baja |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 51: Requisito Software de Funcionalidad del Sistema 016

| RS-F-017 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir programar la incorporación a catálogo de productos nuevos en una fecha determinada | | |
| Verificabilidad | Media | Prioridad | Media |
| Necesidad | Opcional | Fuente | Cliente |

Tabla 52: Requisito Software de Funcionalidad del Sistema 017

| RS-F-018 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir ofrecer de forma individual y durante un periodo de tiempo productos a un precio diferente (oferta) | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Deseable | Fuente | Cliente |

Tabla 53: Requisito Software de Funcionalidad del Sistema 018

| RS-F-019 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe tener la capacidad de gestionar clientes | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 54: Requisito Software de Funcionalidad del Sistema 019

| RS-F-020 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe poder asociar los clientes a diferentes grupos de clientes | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 55: Requisito Software de Funcionalidad del Sistema 020

| RS-F-021 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder asignar en los productos, precios especiales en función del grupo de clientes | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 56: Requisito Software de Funcionalidad del Sistema 021

| RS-F-022 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe tener un sistema de gestión de ventas completo | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 57: Requisito Software de Funcionalidad del Sistema 022

| RS-F-023 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe tener un sistema de contabilidad completo (Órdenes de pedido, facturas y facturas de abono) | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 58: Requisito Software de Funcionalidad del Sistema 023

| RS-F-024 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe generar estadísticas de ventas | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 59: Requisito Software de Funcionalidad del Sistema 024

| RS-F-025 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe tener un sistema de gestión de envíos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 60: Requisito Software de Funcionalidad del Sistema 025

| RS-F-026 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la introducción de diferentes direcciones de facturación y envío | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 61: Requisito Software de Funcionalidad del Sistema 026

| RS-F-027 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la gestión de envíos a través de operadores logísticos | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Media | Fuente | Cliente |

Tabla 62: Requisito Software de Funcionalidad del Sistema 027

| RS-F-028 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema de envíos debe ser automatizable de forma que los operadores logísticos puedan procesar los pedidos listo para su envío de la tienda y efectuar ellos mismos todo el proceso. Todo el proceso debe ser automático | | |
| Verificabilidad | Media | Prioridad | Baja |
| Necesidad | Baja | Fuente | Cliente |

Tabla 63: Requisito Software de Funcionalidad del Sistema 028

| RS-F-029 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir diferentes formas de pago | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 64: Requisito Software de Funcionalidad del Sistema 029

| RS-F-030 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe poder gestionar la introducción de reglas de carro de la compra, de forma que en función del contenido del mismo se puedan aplicar diferentes promociones, tales como envío gratuito o 2x1 | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 65: Requisito Software de Funcionalidad del Sistema 030

| RS-F-031 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar la introducción de reglas de catálogo, de forma que se pueda aplicar de forma masiva descuentos o precios especiales. | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Media | Fuente | Cliente |

Tabla 66: Requisito Software de Funcionalidad del Sistema 031

| RS-F-032 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe poder permitir la creación y gestión de cupones de descuento | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 67: Requisito Software de Funcionalidad del Sistema 032

| RS-F-033 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir la gestión de varias monedas diferentes por tienda. Los precios en otra moneda se mostrarán a modo informativo. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 68: Requisito Software de Funcionalidad del Sistema 033

| RS-F-034 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la elección de varios idiomas diferentes por tienda. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 69: Requisito Software de Funcionalidad del Sistema 034

| RS-F-035 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe actualizar de forma diaria los cambios en las cotizaciones de la moneda, con el fin de realizar conversiones reales | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 70: Requisito Software de Funcionalidad del Sistema 035

| RS-F-036 | | | |
|------------------------|---|------------------|------|
| Descripción | El sistema debe estar preparado para poder realizar una configuración SEO básica de la tienda | | |
| Verificabilidad | Alta | Prioridad | Alta |

| | | | |
|------------------|----------|---------------|---------|
| Necesidad | Esencial | Fuente | Cliente |
|------------------|----------|---------------|---------|

Tabla 71: Requisito Software de Funcionalidad del Sistema 036

| RS-F-037 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder gestionar impuestos (IVA) diferentes por país | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 72: Requisito Software de Funcionalidad del Sistema 037

| RS-F-038 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la creación de diferentes impuestos (IVA) en cada país | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 73: Requisito Software de Funcionalidad del Sistema 038

| RS-F-039 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe poder permitir la implantación de módulos que añadan funcionalidades al sistema | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 74: Requisito Software de Funcionalidad del Sistema 039

| RS-F-040 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe tener una API que permita la interacción con dispositivos externos | | |
| Verificabilidad | Alta | Prioridad | Media |
| Necesidad | Media | Fuente | Cliente |

Tabla 75: Requisito Software de Funcionalidad del Sistema 040

| RS-F-041 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe de realizar envíos de emails en las diferentes fases de compra | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 76: Requisito Software de Funcionalidad del Sistema 041

| RS-F-042 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la personalización de los emails | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 77: Requisito Software de Funcionalidad del Sistema 042

| RS-F-043 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema debe permitir la gestión de roles de permisos en Backoffice [19] | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 78: Requisito Software de Funcionalidad del Sistema 043

| RS-F-044 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe permitir la creación de usuarios y su asignación con roles de permisos de Backoffice | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 79: Requisito Software de Funcionalidad del Sistema 044

3.1.2.3 Requisitos de interfaz del sistema

| RS-I-001 | |
|--------------------|---|
| Descripción | El diseño de la web se debe adaptar automáticamente a las medidas del dispositivo desde el que se esté accediendo, de |

| | | | |
|------------------------|---|------------------|---------|
| | manera que la web sea perfectamente visible y funcional. Es decir, la interfaz debe ser “responsive”. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 80: Requisito Software de Interfaz del Sistema 001

| RS-I-002 | | | |
|------------------------|--|------------------|---------|
| Descripción | Se debe crear un diseño diferente por cada tienda. | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 81: Requisito Software de Interfaz del Sistema 002

3.1.3 Requisitos de restricción

| RR-001 | | | |
|------------------------|---|------------------|---------|
| Descripción | La interfaz “responsive” para PC se testeará con los principales terminales del mercado: PC, Portátil 15”, Portátil 14”, Mac y Mac Book Air | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 82: Requisito de Restricción 001

| RR-002 | | | |
|------------------------|--|------------------|---------|
| Descripción | La interfaz “responsive” para tablet se testeará con los principales terminales del mercado: Galaxy Tab y Ipad | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 83: Requisito de Restricción 002

| RR-003 | | | |
|------------------------|---|------------------|---------|
| Descripción | La interfaz “responsive” para Smartphone se testeará con los siguientes terminales: Galaxy S2, Galaxy S3, Galaxy S4, Galaxy S5, Galaxy S6, Galaxy S6 Edge, Iphone 4, Iphone 4S, Iphone 5, Iphone 5s, Iphone 6 y Iphone 6 Plus | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 84: Requisito de Restricción 003

3.1.4 Requisitos de diseño del sistema hardware

| RD-001 | | | |
|------------------------|---|------------------|---------|
| Descripción | El sistema ha de usar una tecnología de alta disponibilidad que permita alcanzar un 99.5% de disponibilidad anual | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 85: Requisito de Diseño del Sistema Hardware 001

| RD-002 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe ser fácilmente escalable de forma que tenga la capacidad de absorber picos no previstos de demanda | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 86: Requisito de Diseño del Sistema Hardware 002

| RD-003 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe estable, por lo que se deben tener sistemas redundantes de seguridad | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 87: Requisito de Diseño del Sistema Hardware 003

| RD-004 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe de ser de fácil mantenimiento, de forma que no sea necesario un gran equipo para su manejo y gestión | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 88: Requisito de Diseño del Sistema Hardware 004

| RD-005 | | | |
|------------------------|--|------------------|---------|
| Descripción | El sistema debe tener una política de backups que impida la pérdida de datos | | |
| Verificabilidad | Alta | Prioridad | Alta |
| Necesidad | Esencial | Fuente | Cliente |

Tabla 89: Requisito de Diseño del Sistema Hardware 005

3.2 Desarrollo de software

En este punto trataremos los distintos procesos implicados en el desarrollo de un producto software.

3.2.1 Diagrama de casos de uso

3.2.1.1 Nivel 0

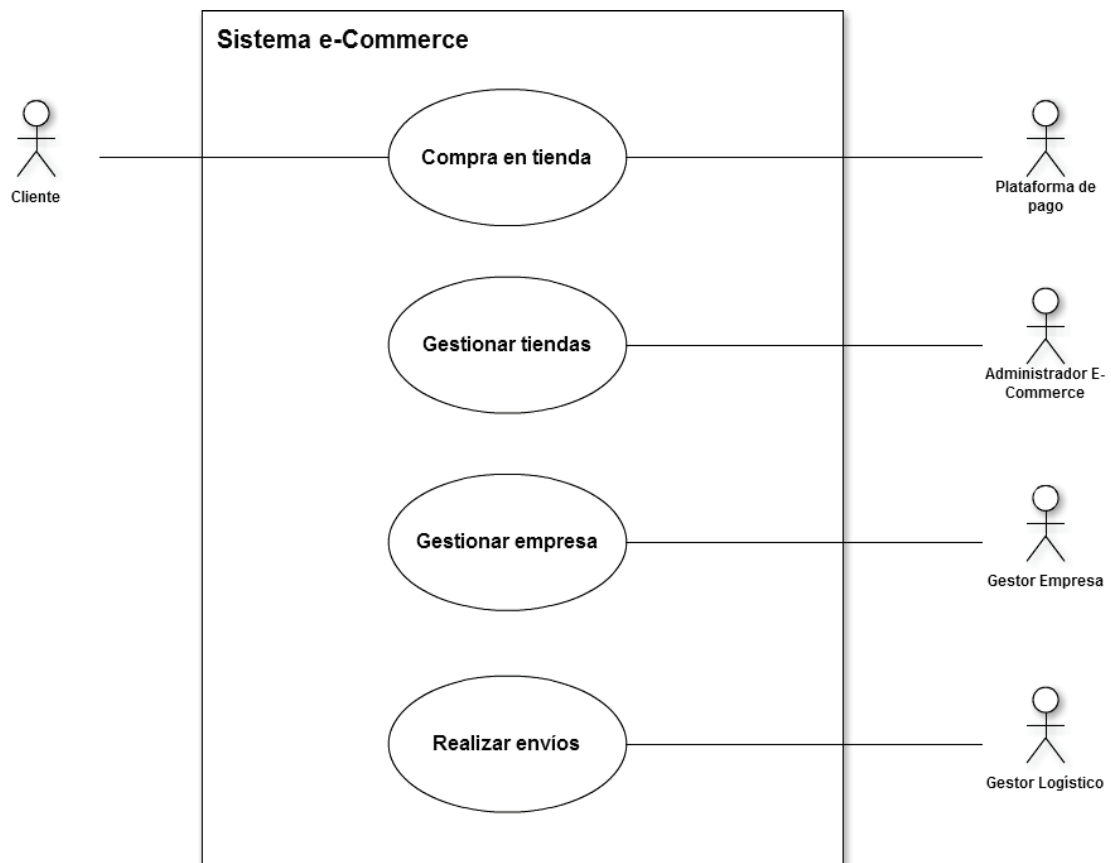


Ilustración 4: Diagrama de casos de uso – Nivel 0

3.2.1.2 Nivel 1

Con el fin de reforzar el primer diagrama, se ha considerado necesario profundizar un nivel en la funcionalidad del caso de uso “Compra en tienda”

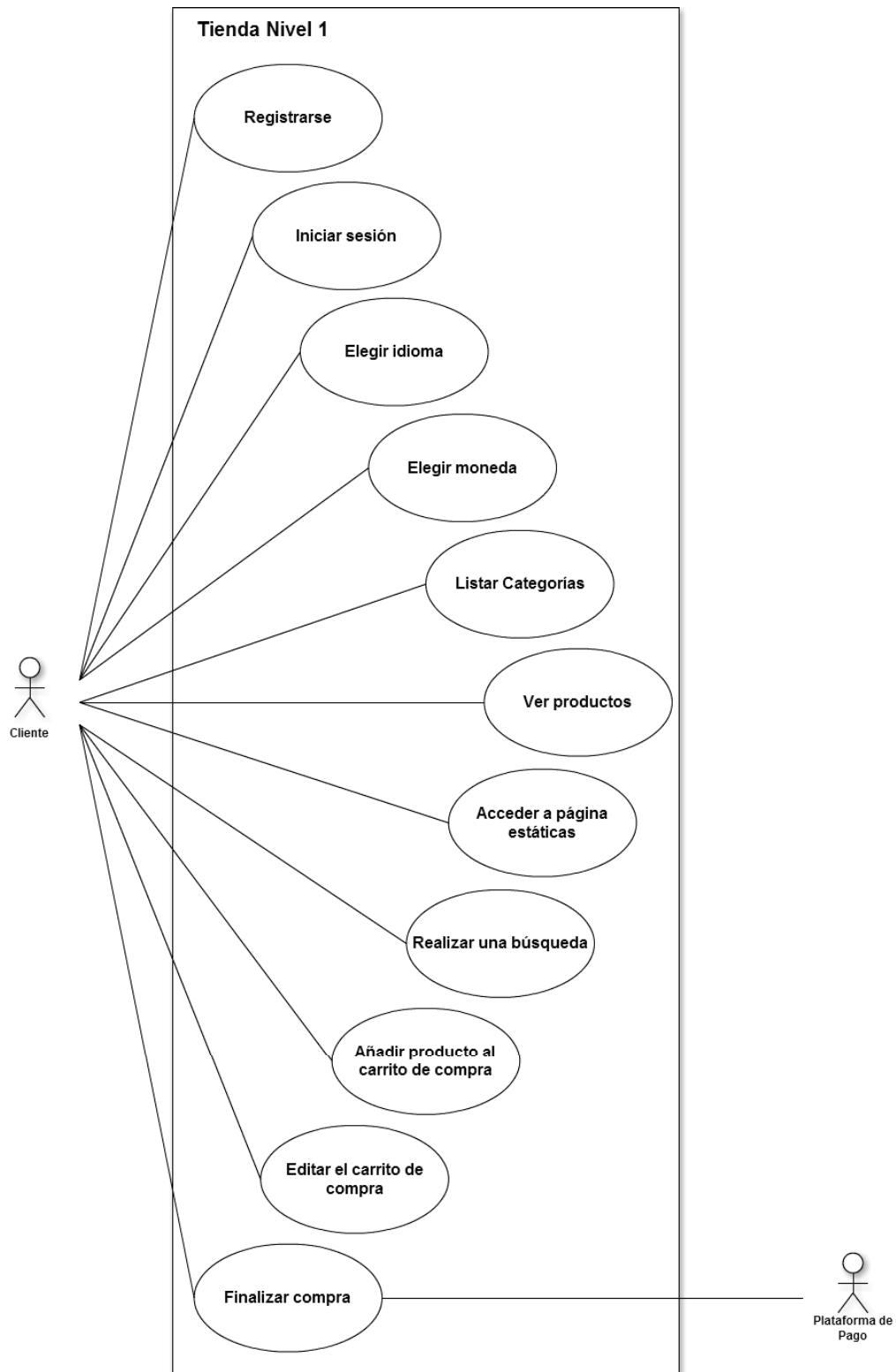


Ilustración 5: Diagrama de casos de uso – Nivel 1

3.2.2 Casos de Uso

3.2.2.1 Descripción

En el siguiente apartado identificaremos los casos de uso del sistema. Para ello nos ayudaremos del siguiente cuadro:

| Identificador de caso de uso | |
|------------------------------|--|
| Descripción | |
| Actor | |
| Pre-Condiciones | |
| Escenario | |
| Post-Condiciones | |

Tabla 90: Cuadro ejemplo casos de uso

Descripción de los campos:

- Identificador: Ha de ser único. Se utilizará la nomenclatura CU-XXX, donde XXX hace referencia al número de caso de uso.
- Descripción: Descripción del caso de uso.
- Actor: Indica el rol que realiza el caso de uso.
- Pre-Condiciones: Condiciones que se deben cumplir para poder realizar el caso de uso.
- Escenario: Pasos que componen el caso de uso
- Post-Condiciones: Estado del sistema una vez realizado el caso de uso

3.2.2.2 Identificación de los Casos de Uso

| CU-001 | |
|-------------|-------------------|
| Descripción | Compra en tienda. |
| Actor | Cliente. |

| | |
|-------------------------|--|
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | Introducir en el navegador la dirección de la tienda de comercio electrónico: https://aws.mgnt.project.com . |
| Post-Condiciones | Realizar uno de los siguientes casos de uso: <ol style="list-style-type: none"> 1. Registrarse. 2. Iniciar sesión. 3. Elegir idioma. 4. Elegir moneda. 5. Listar categorías. 6. Listar productos. 7. Acceder a página estáticas. 8. Realizar una búsqueda. 9. Añadir al carrito de compra. 10. Editar el carrito de compra. 11. Finalizar compra. |

Tabla 91: Caso de uso - 001

| CU-002 | |
|-------------------------|---|
| Descripción | Gestionar tiendas. |
| Actor | Administrador E-Commerce. |
| Pre-Condiciones | <ol style="list-style-type: none"> 1. Conocer el acceso al panel de backoffice. 2. Estar en posesión de un usuario y contraseña válido para las funciones de administración de tiendas. |
| Escenario | <ol style="list-style-type: none"> 1. Acceder al panel de acceso al backoffice. 2. Acceder correctamente al panel backoffice haciendo uso de un usuario y contraseña válidos. |
| Post-Condiciones | El cliente tendrá acceso a los recursos del backoffice necesarios para la gestión de las tiendas. |

Tabla 92: Caso de uso - 002

| CU-003 | |
|------------------------|---|
| Descripción | Gestionar empresa. |
| Actor | Gestor Empresa. |
| Pre-Condiciones | <ol style="list-style-type: none"> 1. Conocer el acceso al panel de backoffice. 2. Estar en posesión de un usuario y contraseña válido para las funciones de administración de tiendas. |

| | |
|-------------------------|---|
| Escenario | <ol style="list-style-type: none"> 1. Acceder al panel de acceso al backoffice. 2. Acceder correctamente al panel backoffice haciendo uso de un usuario y contraseña válidos. |
| Post-Condiciones | El cliente tendrá acceso a los recursos del backoffice necesarios para la gestión de la empresa. |

Tabla 93: Caso de uso - 003

| CU-004 | |
|-------------------------|---|
| Descripción | Realizar envíos. |
| Actor | Gestor Logístico. |
| Pre-Condiciones | <ol style="list-style-type: none"> 1. Haber realizado la configuración necesaria en el sistema del gestor logístico. 2. Haber realizado la configuración necesaria en el backoffice de Magento para permitir el acceso del sistema automatizado del gestor logístico. |
| Escenario | <ol style="list-style-type: none"> 1. Cada hora, el robot del gestor logístico realiza una conexión a Magento. 2. Extrae todos los datos de los pedidos que estén marcados como "pagado". 3. Marca dichos pedidos como "Completo". 4. Inserta el número de seguimiento en pedidos más antiguos que ya haya enviado. |
| Post-Condiciones | Magento enviará un email al cliente informando de que su pedido ha sido enviado y del número de seguimiento. |

Tabla 94: Caso de uso - 004

| CU-005 | |
|-------------------------|--|
| Descripción | El cliente se registra en el sistema de comercio electrónico. |
| Actor | Cliente |
| Pre-Condiciones | No estar registrado previamente |
| Escenario | <ol style="list-style-type: none"> 1. El cliente accede al formulario de registro. 2. El cliente rellena todos los datos obligatorios. 3. El usuario formaliza el registro. |
| Post-Condiciones | El cliente puede acceder al sistema con sus credenciales |

Tabla 95: Caso de uso - 005

| CU-006 | |
|-------------------------|---|
| Descripción | El cliente inicia sesión en el sistema de comercio electrónico. |
| Actor | Cliente |
| Pre-Condiciones | Estar registrado previamente |
| Escenario | <ol style="list-style-type: none"> 1. El cliente accede al formulario de acceso. 2. El usuario introduce un usuario y contraseña válidos. |
| Post-Condiciones | El cliente es redirigido a su zona privada. |

Tabla 96: Caso de uso - 006

| CU-007 | |
|-------------------------|--|
| Descripción | El cliente elige el idioma en el que visualizará la tienda. |
| Actor | Cliente. |
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | <ol style="list-style-type: none"> 1. El cliente selecciona el desplegable de idiomas. 2. El cliente selecciona el idioma elegido. |
| Post-Condiciones | La tienda se visualiza en el idioma elegido por el cliente. |

Tabla 97: Caso de uso - 007

| CU-008 | |
|-------------------------|--|
| Descripción | El cliente elige la moneda en el que visualizará los precios de los productos de la tienda. |
| Actor | Cliente. |
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | <ol style="list-style-type: none"> 1. El cliente selecciona el desplegable de monedas. 2. El cliente selecciona la moneda elegida. |
| Post-Condiciones | Los precios de la tienda se visualizan en la moneda elegida por el cliente. |

Tabla 98: Caso de uso - 008

| CU-009 | |
|-------------------------|--|
| Descripción | El cliente selecciona una categoría para ver su contenido. |
| Actor | El cliente. |
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | El cliente elije una categoría del menú y la selecciona. |
| Post-Condiciones | La tienda debe mostrar los productos pertenecientes a la categoría elegida por el cliente. |

Tabla 99: Caso de uso – 009

| CU-010 | |
|-------------------------|--|
| Descripción | El cliente visualiza un producto en detalle. |
| Actor | Cliente. |
| Pre-Condiciones | El cliente debe haber listado una categoría para poder ver los productos y poder seleccionar uno para ver en detalle. Otra posibilidad es que el usuario se encuentre en la página de inicio y pueda seleccionar alguno de los productos expuestos. |
| Escenario | El usuario selecciona un producto para poder verlo en detalle. |
| Post-Condiciones | La tienda muestra la página del producto con todos los detalles del mismo. |

Tabla 100: Caso de uso - 010

| CU-0011 | |
|-------------------------|--|
| Descripción | El cliente accede a una página estática. |
| Actor | Cliente. |
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | El cliente selecciona cualquiera de los enlaces a una página estática. |
| Post-Condiciones | La tienda muestra la página estática seleccionada por el cliente. |

Tabla 101: Caso de uso - 011

| CU-012 | |
|-------------------------|---|
| Descripción | El cliente realiza una búsqueda. |
| Actor | Cliente. |
| Pre-Condiciones | El cliente debe haber accedido a la tienda. |
| Escenario | El cliente introduce el término a buscar en el cuadro del buscador. |
| Post-Condiciones | La tienda muestra el resultado de la búsqueda. |

Tabla 102: Caso de uso – 012

| CU-013 | |
|-------------------------|---|
| Descripción | Agregar producto simple a la cesta de la compra. |
| Actor | El cliente. |
| Pre-Condiciones | El usuario debe estar en la página del producto, en la página de catálogo o página de inicio. |
| Escenario | El cliente selecciona “Añadir al carro de la compra” en el producto deseado. |
| Post-Condiciones | El producto se añade al carro de la compra |

Tabla 103: Caso de uso - 013

| CU-014 | |
|-------------------------|--|
| Descripción | Agregar producto configurable a la cesta de la compra. |
| Actor | El cliente. |
| Pre-Condiciones | El usuario debe estar en la página del producto. |
| Escenario | El cliente selecciona “Añadir al carro de la compra” en el producto deseado. |
| Post-Condiciones | El producto se añade al carro de la compra. |

Tabla 104: Caso de uso - 014

| CU-015 | |
|-------------------------|---|
| Descripción | El cliente edita el carro de la compra. |
| Actor | Cliente. |
| Pre-Condiciones | El carro de la compra debe contener algún producto. |
| Escenario | El cliente modifica la cantidad de los productos. |
| Post-Condiciones | El carro de la compra se actualiza correctamente. |

Tabla 105: Caso de uso - 015

| CU-016 | |
|-------------------------|---|
| Descripción | El cliente finaliza la compra. |
| Actor | Cliente |
| Pre-Condiciones | El carrito de la compra debe contener al menos un producto. |
| Escenario | <ol style="list-style-type: none"> 1. El cliente debe rellenar todos los datos personales obligatorios. 2. El cliente debe residir en un país al cual la tienda realice envíos. 3. El cliente debe elegir un método de envío. 4. El cliente debe elegir un método de pago. 5. El cliente debe aceptar los términos y condiciones 6. El cliente debe finalizar el pedido. 7. El cliente debe pagar el pedido. |
| Post-Condiciones | El pedido se genera en Magento y se marca como pagado, posibilitando que el sistema de envío automático proceda a su envío. |

Tabla 106: Caso de uso - 016

3.2.3 Diagramas de Secuencia

En este apartado mostraremos dos diagramas de secuencia representativos de dos casos de uso. No se realizan más diagramas por considerarse suficientemente explicada la funcionalidad con ambos casos.

3.2.3.1 Diagrama de Secuencia – Caso de Uso 005 - Registro

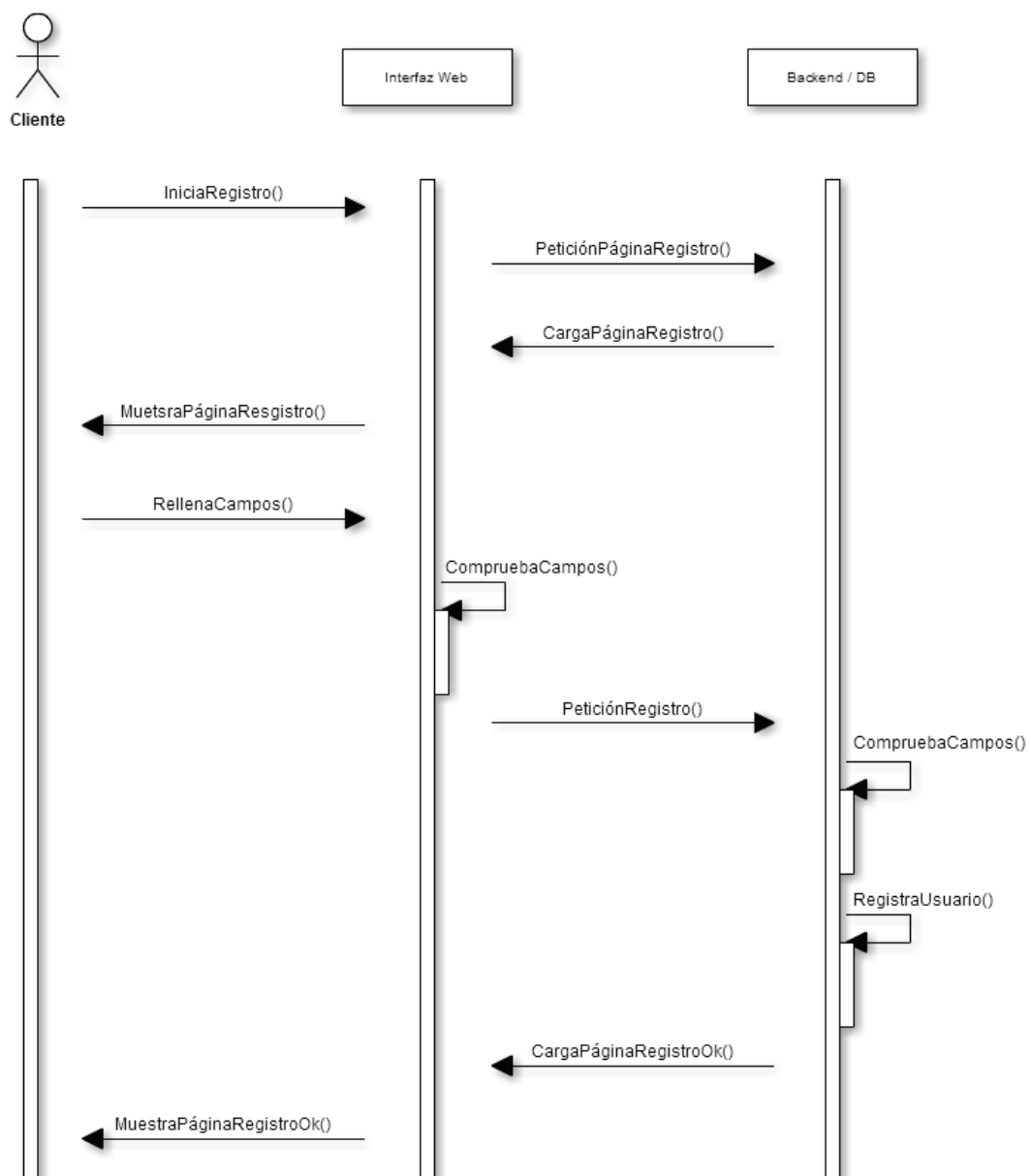


Ilustración 6: Diagrama de Secuencia - Registro

3.2.3.2 Diagrama de Secuencia – Caso de Uso 001 - Compra

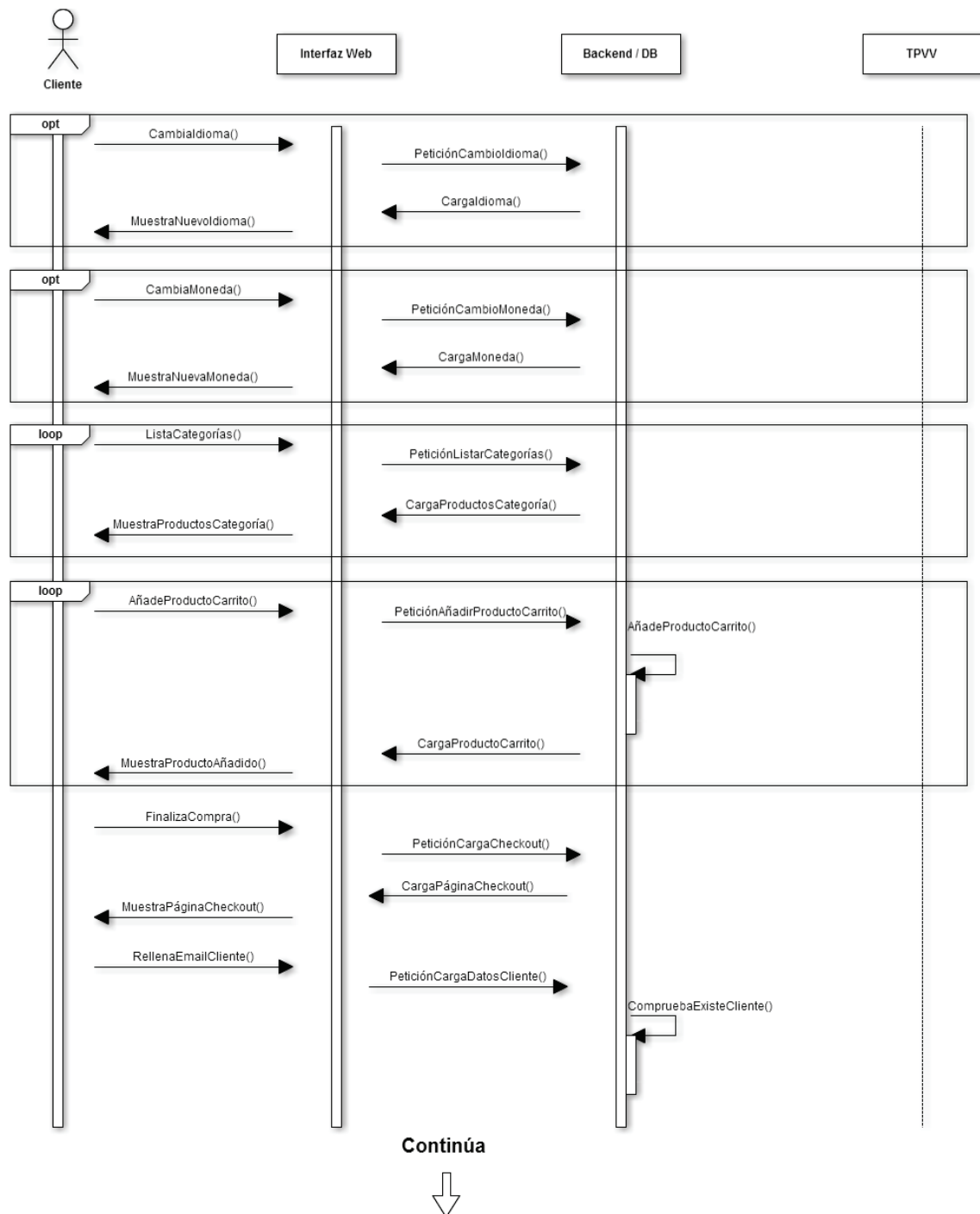


Ilustración 7: Diagrama de secuencia – Compra – Parte1

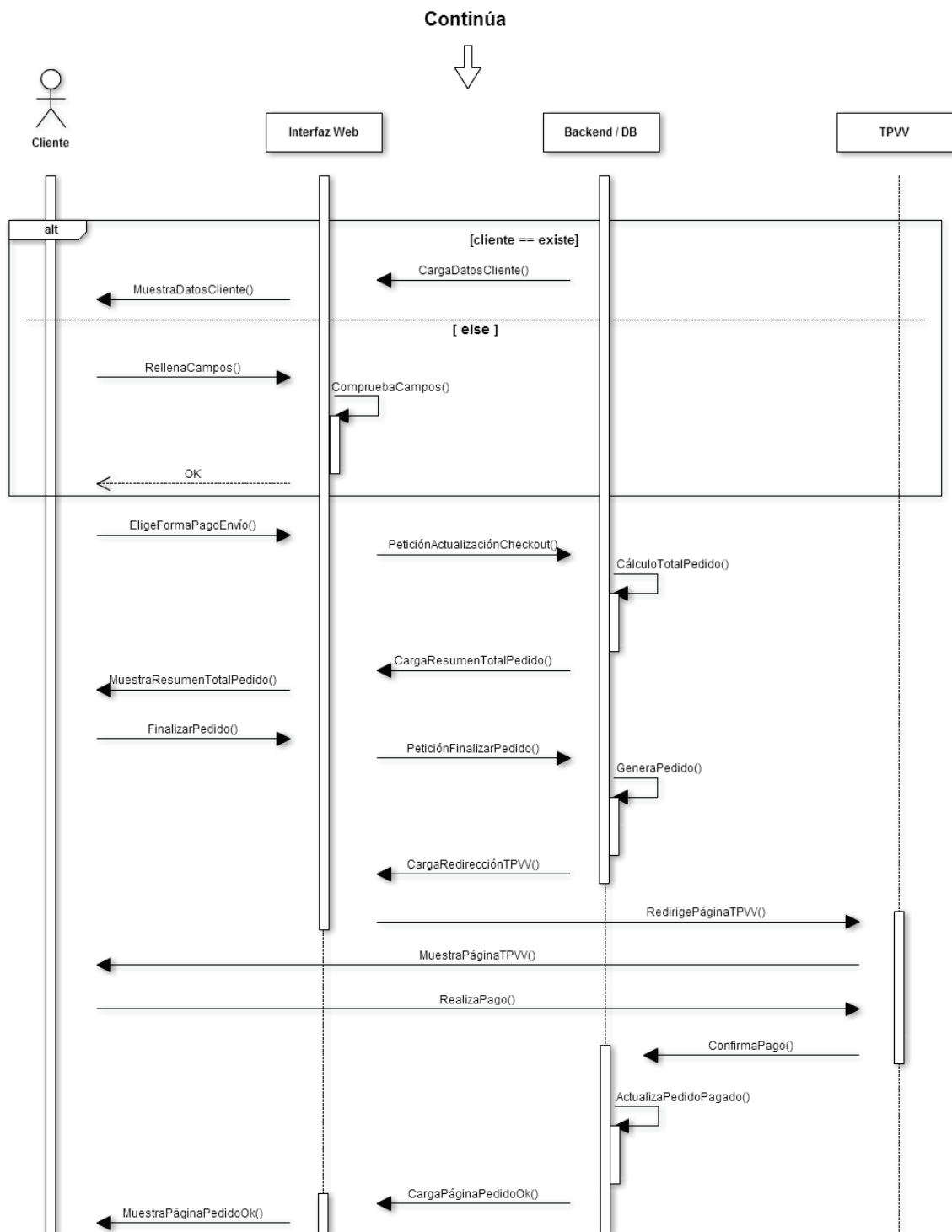


Ilustración 8 Diagrama de secuencia – Compra – Parte2

3.2.4 Diagrama de clases

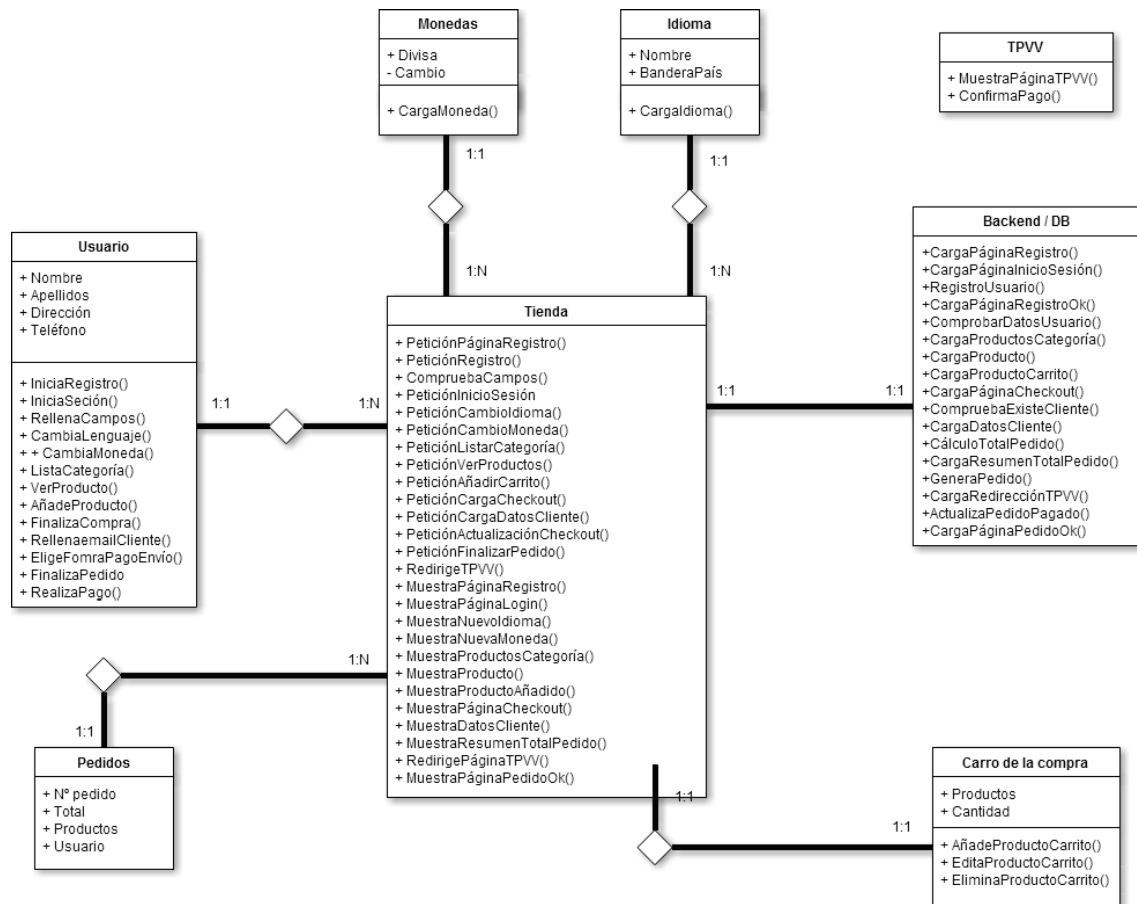


Ilustración 9: Diagrama de Clases

3.3. Infraestructura de Servidor

Tener un servidor expuesto a internet conlleva unos riesgos importantes, máxime si lo que va a alojar el servidor son datos personales de clientes e incluso nuestra propia gestión de la empresa.

Además, es la base de todo nuestro sistema, por lo que hemos de elegir con criterio cada uno de los dos pilares fundamentales en un servidor: hardware y software.

Analizamos todas las opciones existentes:

3.3.1 Sistema Propietario o Sistema Externalizado

El primer paso es la elección de un sistema propio o subcontratado a una empresa externa.

3.3.1.1 Sistema Propietario

El sistema propio tiene permite ser el propietario del hardware, pero conlleva una inversión inicial muy importante, gasto de energía, mantenimiento, gestión de backups (en otro lugar físico), protección ante cortes de luz (UPS), protección ante intrusiones físicas (robo, o manipulación), protección ante desastres naturales (incendios, inundaciones) y una barrera económica grande a la hora de querer aumentar las prestaciones de nuestro equipo, ya que nos obliga a volver a realizar inversión en máquinas nuevas.

3.3.1.2 Sistema Externalizado

El sistema externo nos permite, pagando una determinada cuota o alquiler, olvidarnos de todo lo que no sea la propia gestión del software.

Normalmente nos encontramos con las siguientes tecnologías:

- Hosting [22]: Sistema de recursos compartidos con otros usuarios. Cada empresa ofrece un sistema distinto. Sólo tenemos acceso a nuestra carpeta de usuario donde se nos permite alojar nuestros ficheros/web. Los recursos suelen ser bastante limitados y en la mayor parte de los casos son compartidos con otros usuarios. Muchas empresas usan software como CloudLinux [20] para asignar y asegurar recursos a cada usuario, aunque se sigue compartiendo el sistema. No nos tenemos que preocupar por su mantenimiento. La gestión se suele hacer desde un panel propio de cada empresa o un Cpanel/Plesk [21] limitado.
- VPS [23]: Servidor privado virtual. Se virtualiza una máquina de la que somos los únicos usuarios y a la que se le asignan los recursos específicos contratados. Se

comparte hardware con otros VPS, pero son sistemas contenedores distintos, por lo que nuestras aplicaciones están aisladas de las del vecino. Tiene las desventajas de tener que ser administrado por uno mismo, tanto en lo referente a la instalación/gestión como a las actualizaciones y parches de seguridad. Son más potentes que los hostings tanto por que los recursos disponibles suelen ser mayores como por que no compartimos el sistema con nadie más. Cada vez más empresas incluyen en el precio la administración de estos sistemas.

- Dedicado [24]: Es lo más parecido a un servidor físico. Se contrata hardware concreto y del que somos los únicos usuarios.

En nuestro caso, al querer montar un sistema Magento, de entrada debemos descartar el sistema Hosting, dado los limitados recursos, la poca capacidad de gestión del servidor y la dependencia que tenemos del resto de “vecinos” dentro del hosting.

La elección entre VPS y servidor dedicado es en buena parte económica. Un servidor dedicado es bastante más caro que un VPS al tener a nuestra total disposición unos determinados recursos hardware. Por el contrario, no nos permite aumentar o disminuir recursos en función de nuestras necesidades, con lo que nos encontramos con un sistema con muchas similitudes a la compra de hardware propio.

Por lo tanto, elegimos la opción del VPS.

Diagrama resumen “Infraestructura de servidor”:

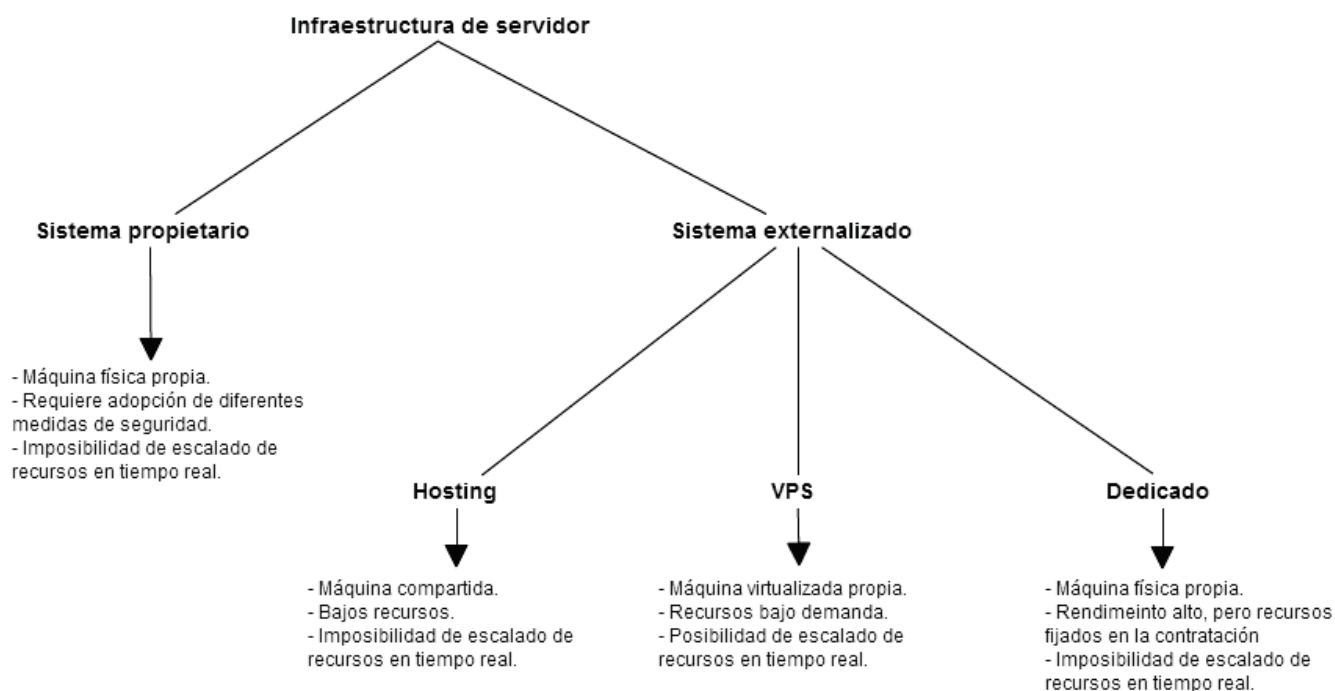


Ilustración 10: Diagrama resumen “Infraestructura de Servidor”

3.3.2 Servidor VPS

3.3.2.1 Tradicional vs Cloud

Con el servidor VPS tradicional, tenemos una máquina con una serie de recursos asignados que, ante la necesidad de aumentarlos, suele ser necesario cambiar de máquina, lo que nos obliga a reinstalar todo el sistema. Es escalable y mucha mejor opción que una máquina propietaria, pero sigue presentando inconvenientes.

En los últimos años ha surgido la informática “en la nube” (Dropbox, iCloud, etc). Esta tendencia no sólo se ha limitado al almacenamiento, sino que también ha llegado al área de los servidores. Los sistemas Cloud [25] permiten iniciar servidores en minutos, con diversas configuraciones e, incluso, aumento de recursos del sistema con el simple inconveniente de tener que apagar la máquina mientras se reasignan los recursos, pero no implica ningún tipo de reinstalación.

A la hora de contratar este tipo de soluciones hay que tener cuidado y leer bien las especificaciones, ya que, dependiendo del proveedor, esta tecnología se comercializa de forma diferente.

Hay sistemas que permiten el aumento de recursos de forma individualizada (vCore [26], RAM [27], o tamaño del disco), pero otros sólo permiten escalar “por niveles” de forma que aumentar un nivel implica aumentar todos los recursos a la vez, lo cual puede suponer un incremento de precio de forma innecesaria. El mayor inconveniente de este tipo de soluciones es si, además, este cambio de nivel implica una migración del sistema a una máquina nueva, con lo que nos encontraríamos ante un sistema tradicional etiquetado como Cloud.

La filosofía Cloud nos ofrece toda una serie de servicios de los cuales se nos permite hacer uso a demanda y en la medida que los necesitemos, pagando siempre por lo que se haya usado y en la medida en la que se haya hecho.

3.3.2.2 Escalado VPS

Hasta el momento hemos ido descartando y eligiendo la opción de servidor que nos fuera los más económica posible pero que a la vez fuera versátil a la hora de incrementar recursos. Ahora tenemos que ver cómo afrontamos ese incremento.

Normalmente, lo lógico es hacer un estudio de las necesidades de nuestro sistema, el número de usuarios concurrentes y el tráfico que van a generar para así poder estimar la carga de nuestro servidor.

Si todo va bien, al incrementarse las visitas/ventas la carga media de nuestro servidor aumente, por lo que deberemos incrementar los recursos para evitar lentitud o el colapso del sistema.

3.3.2.2.1 Escalado vertical

Se define como el incremento de recursos hardware, es decir, contratamos más CPU, RAM, Disco, etc

En este caso según el proveedor existen dos opciones:

- Sistema tradicional: deberemos contratar el siguiente paquete o nivel, teniéndonos que adaptar a las especificaciones que nos den, lo cual, probablemente provoque que paguemos por recursos de los que tengamos necesidad. Suele requerir migración del sistema, al tener que cambiar de máquina.
- Sistema “Cloud”: en este caso se suele poder elegir los recursos a aumentar y, previo apagado de la máquina, los tendremos disponibles.

Este sistema tiene inconvenientes:

- Requiere parada del sistema e, incluso, migración.
- Dependemos de los límites de nuestro proveedor, tanto en el caso de escalar por “paquetes” como por recursos independientes.
- El escalado es permanente, es decir, siempre se está pagando lo mismo uses o no los recursos.

Este tipo de estrategia podría ser viable para pequeñas webs o sistemas internos, pero cuando estamos hablando de una tienda online, ésta debe estar disponible sin interrupciones, por lo que apagar un servidor para aumentar recursos no debería ser una opción.

3.3.2.2.2 Escalado horizontal

Se define como el incremento de recursos basado en el reparto de carga entre varios nodos[28], no en aumentar los recursos de un nodo en concreto. De esta manera, en lugar de aumentar la capacidad de procesamiento o su memoria, creamos distintos servidores idénticos y repartimos la carga ellos.

Esta técnica nos obliga a cambiar el diseño de nuestro sistema, ya que tenemos que tener en distintos nodos el servidor web y la base de datos.

De esta forma podemos crear servidores con recursos y configuraciones más específicos para cada sistema. Por ejemplo, un servidor de bases de datos hará un uso mucho mayor de los recursos de entrada/salida a disco que un servidor web.

Esta configuración es mucho más eficiente y, aunque partimos de un mínimo de dos nodos (web y base de datos, lo cual implica mayor coste), al repartir la carga y ser cada uno más eficiente en la tarea asignada, permite absorber una mayor demanda.

Por otro lado, el escalado horizontal aporta complejidades añadidas al servidor web y a la base de datos.

Para poder realizar este tipo de escalado se hacen imprescindibles una serie de configuraciones que un proveedor de servicios de hosting al uso no es capaz de soportar. De ello hablaremos más adelante.

3.3.2.2.2.1 Escalado horizontal en servidor web

El escalado horizontal en un servidor web se realiza replicando el mismo tantas veces como sea necesario. El acceso al contenido se realiza mediante un balanceador de carga, que es que el decide a qué instancia envía la petición recibida.

Cada servidor web tiene instalada una copia del sistema (en nuestro caso Magento) configurado para acceder a la misma base de datos, por lo tanto, sea cual sea el servidor web al que sea dirigido un usuario, podrá ver la misma tienda.

Este sistema genera diversos problemas, pero nos vamos a ceñir a los concretos que nos podemos encontrar en nuestro caso:

- Control sesiones.
- Acceso a la caché.
- Acceso a las imágenes.
- Deployment / Despliegue[29]

3.3.2.2.2.1.1 Control de sesiones

Como cualquier sistema web, Magento hace uso de cookies para poder hacer un seguimiento del usuario y mostrarle el contenido adecuado. En el momento en el que se accede por primera vez, se inserta una cookie (con un tiempo de vida limitado) en el navegador usado por el potencial cliente y otra guardada en el servidor. De esta forma, Magento puede asociar una cookie con un carro de la compra, con un usuario concreto y con opciones tales como si ha iniciado o no sesión. Es lo que se conoce como una cookie de sesión.

El problema se plantea en el momento en el que tenemos diferentes instancias de servidores web. Aunque los balanceadores de carga suelen tener disponible la opción de asociar mediante sus propias cookies de sesión un cliente con un nodo en concreto, no siempre se puede cumplir. Si no es redirigido al mismo nodo, un cliente que estuviera comprando perdería su carro de la compra, su sesión, etc.

3.3.2.2.1.2 Acceso caché

Magento permite la configuración de varios niveles de caché. El básico, embebido en Magento, consiste en una caché almacenada en disco. Para que el siguiente nivel sea operativo, requiere la instalación de un sistema de caché externo a Magento pero integrado con él, como puede ser APC[30], Memcached[31], Redis[32] o Varnish[33]. La integración se hace efectiva por la configuración directa que permite Magento y por el uso de módulos que permiten la invalidación de las cachés en el momento en el que sea necesario.

Ante el primer nivel de caché, el que se realiza a nivel de sistema de archivos en el propio Magento, nos surge un problema similar que con las sesiones, ya que cada nodo tendría la caché almacenada localmente, lo que la haría inútil al no poder ser actualizada.

3.3.2.2.1.3 Acceso a imágenes

Magento almacena por defecto las imágenes en su sistema de archivos. En caso de modificar la imagen de un producto, crear un producto nuevo, o una acción similar, habría que replicar dicha imagen en todos los nodos para que estuviera disponible en todos ellos.

La solución a los problemas de caché, imágenes y sesiones de usuario se encuentra en el almacenamiento compartido. Un servidor NFS[34] (Network File System) se encargará de compartir en la red una carpeta para la imágenes, otra para los archivos de sesiones y otra para los archivos de caché. Todos los nodos configurarán las rutas de sesión, imágenes y caché de forma que se conecten al servidor NFS. De esta forma, en el momento en el que un cliente acceda a un nodo, éste generará el archivo de sesión que será accesible desde todos los nodos. Lo mismo ocurriría para las imágenes y los archivos de caché.

3.3.2.2.1.4 Deployment / despliegue

Desde el momento en el que tenemos varios nodos idénticos ejecutando el mismo código, se plantea un reto a la hora de la actualización del mismo ya que, siendo el número que sean, no debemos optar por la actualización del código nodo a nodo (no es lo mismo tener que actualizar 5 nodos, que 50). Esta tarea se debe automatizar de forma que se haga simultáneamente y en mínimo periodo de tiempo posible.

Una opción es el uso de programas de “Deployment” o “Despliegue”, pensados no sólo para un primer despliegue, sino también para ejecutar tareas de actualización de código de forma masiva.

El esquema de trabajo sería conectar los nodos al servidor de despliegue para que, cada cierto tiempo o ante un evento, actualicen el código.

Ejemplos de estas herramientas son Cheff[35], Puppet[36] o Jenkins[37].

En nuestro caso concreto nos podríamos encontrar ante la necesidad de instalar un nuevo módulo en nuestro sistema Magento, para lo cual necesitaríamos realizar el despliegue en todos los nodos activos y a la vez.

Sería un error hacer uso del sistema de instalación de módulos del que viene provisto este CMS, ya que entonces sólo sería instalado en el nodo en el que estemos accediendo, pero no en el resto de ellos que estuvieran disponibles. Además, esta acción podría generar diversos problemas que dañarían nuestro sistema. Por ejemplo la base de datos tendría información relativa a ese módulo, como su estado, pero si accedemos desde un nodo que no tenga dicho código instalado, podría generar un error fatal.

3.3.2.2.2 Escalado horizontal en base de datos

Escalar horizontalmente un Sistema Gestor de Bases de Datos Relacional como MySQL[38] puede llegar a ser bastante problemático más aún si, como en nuestro caso, hacemos uso del motor InnoDB[39], que ejecuta transacciones ACID[40] (Atomicity, Consistency, Isolation and Durability).

Vamos a ver diferentes métodos de escalado:

3.3.2.2.2.1 Escalado horizontal en base de datos: Master/Slave

Es el método más común. La base de datos principal se le asigna el rol principal o “Master” y se crea una réplica esclava o “Slave” de sólo lectura. Siguiendo los principios de las transacciones ACID, una transacción no se considerará finalizada hasta que no se haya ejecutado la operación en la base de datos “maestra” y “esclava”, por lo tanto, la réplica de lectura siempre estará actualizada y podrá ser usada para operaciones de lectura y/o backup, descargando la base de datos “Maestra” y, por tanto, escalando horizontalmente el SGBD sin comprometer los datos.

Ventajas:

- ✓ Técnica sencilla.
- ✓ Tecnología bien implementada y controlada.
- ✓ Permite balanceo de consultas SELECT
- ✓ Permite gestión de backups sin sobrecargar la base de datos principal.

- ✓ El nodo “slave” puede tener asignados menos recursos al sólo recibir peticiones de lectura. De esta forma se mejora la capacidad del SGBD sin duplicar costes.

Desventajas:

- ✗ En caso de fallo de la base de datos principal, la réplica de lectura no tiene más uso que de lectura. Se provocaría una desincronización. Habría que intervenir para repararla una vez la base de datos principal estuviera en línea o cambiar su configuración de forma que la réplica pasase a convertirse en “Maestra” (failover).
- ✗ Requiere configuración avanzada del SGDB
- ✗ Mayor coste al hacer uso de otra instancia.

3.3.2.2.2.2 Escalado horizontal en base de datos: Master/Master

Deriva de la anterior técnica y consiste en ejecutar dos SGDB con posibilidad de modificación pero siempre haciendo uso de transacciones ACID. De esta forma se dispone de dos nodos sobre los que balancear todas las peticiones.

Ventajas:

- ✓ Ante una gran cantidad de peticiones de escritura permite absorber mejor la demanda al balancear entre dos nodos las peticiones.
- ✓ En caso de fallo de un nodo, el restante tiene capacidad para mantener el sistema en funcionamiento (no como en el caso anterior, que estaríamos supeditados a peticiones de sólo lectura).

Desventajas:

- ✗ Mayor probabilidad de que se rompa la sincronización entre nodos, lo que implicaría una parada del sistema y un proceso costoso de reparación.
- ✗ Requiere configuración avanzada del SGDB
- ✗ Mayor coste que el caso anterior.

3.3.2.2.2.3 Escalado horizontal en base de datos: Particionado funcional

Técnica consistente en derivar determinadas peticiones (funcionalidades) del programa a otro nodo SGBD. De esta forma, se puede decir que se divide la base de datos y, dependiendo de lo que se quiera consultar, la consulta será dirigida a uno u otro nodo.

Ventajas:

- ✓ Se mejora la capacidad del SGBD al no sólo discernir entre peticiones de escritura / lectura, si no poder dividir realmente la carga al poder repartir la peticiones más costosas entre las distintos nodos.

Desventajas:

- ✗ Gran coste de configuración y gestión.
- ✗ Si por cualquier motivo se acaba saturando la base de datos, cambiar la forma de trabajo es difícil.

3.3.2.2.2.4 Escalado horizontal en base de datos: database sharding

Parecido a la opción anterior. En este caso se diferencia por algún tipo de clave, por ejemplo, DNI. Si el DNI acaba en un número par, se guarda en un nodo y si no en el otro.

Ventajas:

- ✓ Se mejora la capacidad del SGBD al repartir la carga entre dos nodos, permitiéndose la lectura / escritura en ambos.

Desventajas:

- ✗ Gran coste de configuración y gestión.
- ✗ Menor rendimiento que con el particional funcional
- ✗ Si por cualquier motivo se acaba saturando la base de datos, cambiar la forma de trabajo es difícil.

Diagrama resumen “Escalado en servidor VPS”:

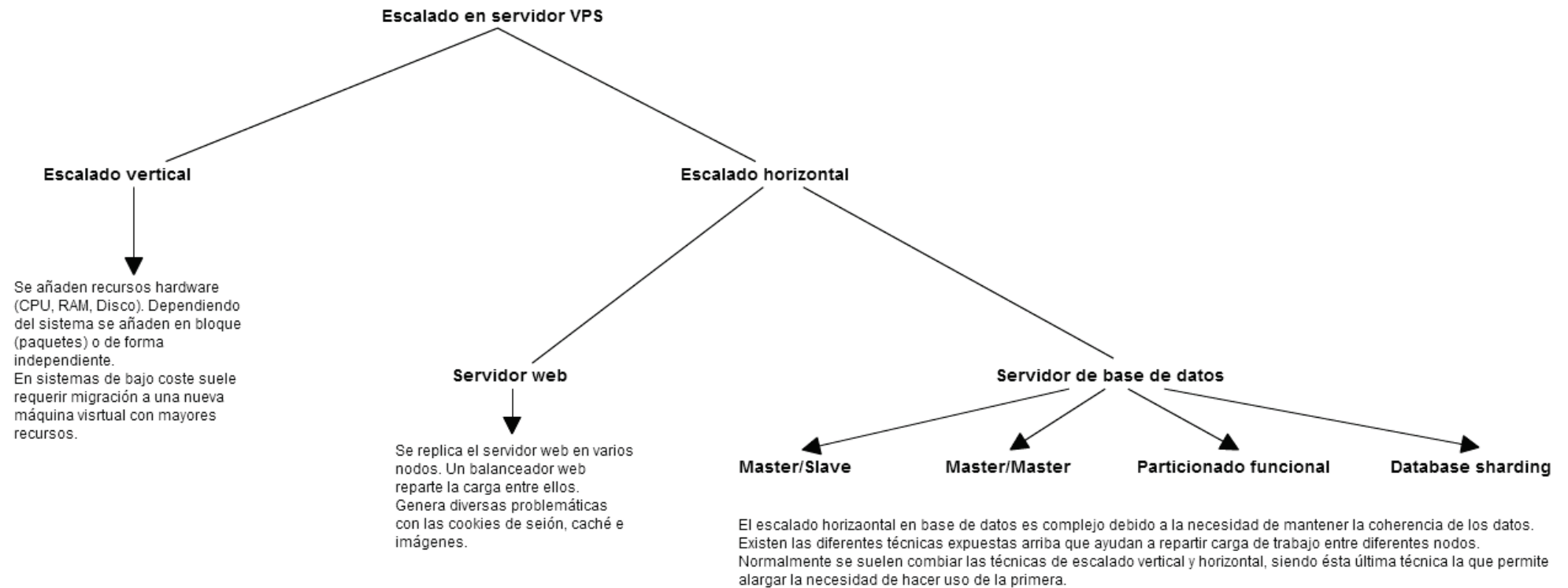


Ilustración 11: Diagrama resumen "Escalado en servidor VPS"

3.3.3 Software

3.3.3.1 Sistema Operativo

Ante la elección de un sistema operativo u otro, una de las variables a evaluar es si se requiere un sistema de software libre o bajo licencia, distinguiendo de esta forma dos grandes ramas: Microsoft Windows[41] y las distribuciones Linux GPL[42]

Uno de los objetivos de este proyecto es el uso de software libre, ya no sólo por el ahorro económico sino por el espíritu de colaboración que desemboca en un continuo desarrollo de este tipo de sistemas, tanto a nivel de funcionalidad como de desarrollo. Por tanto la elección del sistema operativo será una distribución Linux.

Existen numerosas distribuciones, aunque se pueden destacar las dos más grandes e influyentes en entornos web, las cuales son Debian[43] y CentOS[44], siendo esta última más cercana al entorno empresarial por derivar de RedHat[45].

Ambas distribuciones son famosas por su estabilidad, debido a que los nuevos desarrollos superan varios meses en pruebas. Ello provoca que reciban las actualizaciones y últimas versiones de los paquetes de software bastante más tarde que otras distribuciones.

Un gran cantidad de distribuciones están basadas en Debian, por lo que por regla general, es más sencillo encontrar documentación y comunidad de soporte. Además, en general tiende a estar ligeramente por delante de CentOS en lo que a recibir nuevas versiones de software. Ello unido a su mayor facilidad para su actualización a nuevas versiones, hace que sea el sistema operativo elegido.

Debian GNU / Linux 8, también conocido como Jessie, es la versión más estable en este momento de este sistema operativo cuyo lanzamiento inicial se realizó en 1993. Se rige por la licencia GPL, implementa el gestor de paquetes dpkg[46] y como gestor de actualizaciones por defecto usa APT[47] (Advanced Packaging Tool).

3.3.3.2 Servidor Web

El servidor HTTP Apache es un servidor web HTTP de código abierto, para plataformas Unix (BSD, GNU/Linux, etc.), Microsoft Windows, Macintosh y otras, que implementa el protocolo HTTP/1.12 y la noción de sitio virtual. Cuando comenzó su desarrollo en 1995 se basó inicialmente en código del popular NCSA HTTPd 1.3, pero más tarde fue reescrito por completo. [1-bib]

El servidor Apache se desarrolla dentro del proyecto HTTP Server (httpd) de la Apache Software Foundation[48].

Apache presenta entre otras características altamente configurables, bases de datos de autenticación y negociado de contenido, pero fue criticado por la falta de una interfaz gráfica que ayude en su configuración.

Apache tiene amplia aceptación en la red: desde 1996, Apache, es el servidor HTTP más usado. Alcanzó su máxima cuota de mercado en 2005 siendo el servidor empleado en el 70% de los sitios web en el mundo, sin embargo ha sufrido un descenso en su cuota de mercado en los últimos años.

Al ser un software modular, se le pueden aplicar diferentes características activándolas según las necesidades, aligerando su peso al no tener activas todas aquellas funcionalidades que no sean necesarias.

3.3.3.3 PHP

PHP es un lenguaje de programación de uso general de código del lado del servidor originalmente diseñado para el desarrollo web de contenido dinámico. Fue uno de los primeros lenguajes de programación del lado del servidor que se podían incorporar directamente en el documento HTML en lugar de llamar a un archivo externo que procese los datos.

El código es interpretado por un servidor web con un módulo de procesador de PHP que genera la página Web resultante. PHP ha evolucionado por lo que ahora incluye también una interfaz de línea de comandos que puede ser usada en aplicaciones gráficas independientes. Puede ser usado en la mayoría de los servidores web al igual que en casi todos los sistemas operativos y plataformas sin ningún costo. [2-bib]

El uso de PHP se regula bajo la Licencia de PHP[49]

3.3.3.4 Sistema Gestor de Bases de Datos (SGBD)

Una base de datos es un conjunto de información que contiene datos sobre diversos temas pero que comparten una relación entre sí. Puede contener un simple listado de artículos o grandes cantidades de información relativa a una gran empresa. [3-bib]

Un sistema gestor de base de datos es un conjunto de programas que permiten el almacenamiento, modificación y extracción de la información en una base de datos, además de proporcionar herramientas para añadir, borrar, modificar y analizar los datos. Los usuarios pueden acceder a la información usando herramientas específicas de interrogación y de generación de informes, o bien mediante aplicaciones al efecto. [4-bib]

Estos sistemas también proporcionan métodos para mantener la integridad de los datos, para administrar el acceso de usuarios a los datos y para recuperar la información si el sistema se corrompe.

3.3.3.4.1 MySQL

MySQL es un sistema de gestión de bases de datos relacional, multihilo, multiusuario y de código abierto (Licencia GPL).

Una base de datos relacional se caracteriza por contener la información en tablas y las relaciones entre los datos deben estar contenidas en ellas

3.4 Proveedor de Servicios

En apartados anteriores hemos estado evaluando las diferentes tecnologías que existen y sus diferentes formas de escalado. El siguiente paso es la elección del proveedor de servicios que utilizaremos.

Algunos de los objetivos del proyecto, como tener un sistema auto escalable, con un alto nivel de servicio (SLA[50], Acuerdo de Nivel de Servicio) y con capacidad para cubrir prácticamente todas las necesidades asociadas a la implementación del sistema, requieren un proveedor de servicios que nos ofrezca un modelo IaaS[51] (Infraestructure as a Service o Infraestructura como Servicio).

Este es el modelo típico cuando se habla de Sistemas Cloud. Tenemos a nuestra disposición una gran variedad de servicios que podemos contratar, configurar y pagar en función de su uso, tanto en el tiempo como en recursos aprovisionados.

Debemos tener en cuenta la legislación del país donde esté ubicada la empresa ya que habrá que cumplir una gran cantidad de normativas, sobre todo en lo referente a la Protección de Datos, la cual, tiene en cuenta la ubicación del servidor. Nuestro proveedor tiene que permitirnos la ubicación de los servidores en zonas acordes con nuestra legislación.

Además, la ubicación geográfica del servidor puede ser muy importante ya que influye la distancia a la que esté del cliente. En nuestro caso, al estar la tienda destinada a clientes en todo el mundo, es menos relevante.

Existen diversos proveedores (Google Cloud[52], Microsoft Azure[53], Rackspace[54], etc) que cumplen con todos o la mayor parte de los puntos anteriormente indicados. De todos ellos, hemos elegido Amazon Web Services[55]. Algunas razones son: Mayor propagación a nivel mundial, facilidad de uso (en comparación con competidores), modelo de pago por uso y buena documentación (tanto propia como en la comunidad).

Aunque todos los proveedores ofrecen diferentes niveles de soporte, en este tipo de caso siempre hay que dejar este tipo de sistemas en manos de un SysAdmin que nos revise el correcto funcionamiento y mantenga nuestro sistema actualizado y seguro.

3.4.1 Amazon Web Services

Amazon Web Services (en adelante AWS) es una colección de servicios de computación en la nube (también llamados servicios web) que en conjunto forman una plataforma de computación en la nube.

Dicha colección de servicios permite tener en un solo proveedor todo lo necesario para crear una nube privada, con alta disponibilidad y alto rendimiento gracias a su tecnología de auto-escalado.

AWS dispone de varias regiones (Oregon, Irlanda, Sao Paulo, etc) que permiten ser configuradas de forma independiente. Dentro de cada región existen varias zonas de disponibilidad, que no son más que diferentes CPD en distintas ubicaciones dentro de la misma región e interconectados entre sí.

AWS usa una versión modificada de Xen [56], monitor de máquina virtual de código abierto. Dispone de los dos tipos de virtualización soportada por Xen, paravirtualización [57] (PV) y virtualización completa [58] (HVM). En los últimos meses está relegando la paravirtualización en favor de la virtualización completa, mucho más eficiente.

A la hora de la elección del hardware, AWS permite configurar el aprovisionamiento de las instancias de una forma mucho más flexible que otras plataformas. Se puede elegir entre una gran variedad de tipos instancias, algunas de ellas con finalidades específicas tales como computación, uso de memoria RAM, uso de red, uso de CPU o un término medio entre todas ellas. Dentro de cada tipo hay diversos niveles de aprovisionamiento hardware.

Otro aspectos que diferencia a AWS de sus competidores, es el hecho de poder configurar el sistema de almacenamiento de nuestra instancia de forma que no sólo se puede elegir su tamaño, si no el tipo de disco e incluso, si la elección es un disco SSD, su rendimiento (IOPS). Además se nos permite cifrarlo, añadiendo así una capa más de seguridad a nuestros datos.

AWS dispone de una API pública para cada uno de sus servicios, permitiendo de esta manera infinidad de posibilidades a la hora de crear y controlar sus servicios. Para realizar de forma más sencilla las tareas de creación de los sistemas, posee una interfaz web que permite la creación gestión de toda su infraestructura de una manera sencilla y potente.

Recientemente la unión Europea ha aprobado el acuerdo de procesamiento de datos en AWS [59] (APD), un paso más que certifica a AWS como proveedor que protege adecuadamente los datos almacenados en él.

Glosario de términos de AWS que usaremos:

- EC2 (Amazon Elastic Compute Cloud) [62]: Máquina virtual en la nube de AWS.
- AMI (Amazon Machine Image) [61]: Imagen con Sistema Operativo usada por AWS para ejecutar en las instancias EC2.
- EBS (Elastic Block Storage) [63]: Servicio de almacenamiento (volátil) de EC2.
- IOPS: Operaciones de Entrada/Salida por segundo.
- RDS (Amazon Relational Database) [64]: Servicio de Base de datos relacional en la nube de AWS.
- Region [65]: Región del planeta donde AWS permite el despliegue de sus servicios.
- Zona de disponibilidad [66]: Cada Región se compone de varias zonas de disponibilidad interconectadas por redes de muy baja latencia.
- Multi-AZ [67]: Opción que ofrece una alta disponibilidad (99.95% monthly up time SLA) de un servicio a través de una o varias réplicas ubicadas en diferentes zonas de

disponibilidad y gestionadas por Amazon. En el caso concreto de RDS, gestiona un modelo Master-Slave en diferentes zonas de disponibilidad con failover [68] y failback [69] automático.

- CloudWatch [70]: Servicio de supervisión de los recursos en la nube de AWS
- Grupo de Seguridad [71]: Conjunto de reglas de firewall que definen el acceso a la red de los nodos asociados al Grupo.
- S3 (Simple Storage Service) [72]: Servicio de almacenamiento en la nube de AWS.
- CloudFront [73]: Servicio CDN (Content Delivery Network) de AWS.
- Autoscalling [74]: Servicio de autoescalado de recursos EC2 de AWS. Para la monitorización de los recursos se usa CloudWatch.
- Elastic Load Balancer [75]: Servicio de balanceadores de carga de AWS.
- Elastic IP [76]: Servicio de IP estática de AWS.
- VPC (Virtual Private Cloud) [77]: Servicio de nube privada de AWS.
- VPC-Subnet [78]: Subred dentro de una nube privada de AWS. Las subredes se asocian a distintas zonas de disponibilidad.
- IAM [79]: Identity and Access Management, permite controlar de forma segura el acceso a servicios y recursos de AWS por parte de sus usuarios.
- Internet Gateway [80]: Componente dentro de una VPC que permite la comunicación de dicha VPC con Internet.
- Router [81]: Permite definir las reglas de enrutado dentro de una VPC. De esta forma podemos elegir quién se puede comunicar con quién a nivel interno o quien tiene acceso o es accesible desde internet.
- Route53 [82]: Servicio de AWS que permite la gestión de nombres de dominio y DNS [83].

Usando las distintas tecnologías disponibles en AWS, se han elaborado distintos diseños para ver las ventajas de cada uno de ellos y estudiar cómo aplicar lo visto anteriormente en cuanto a las distintas técnicas de escalado.

Una vez explicados los conceptos básicos y servicios que nos proporciona AWS, pasamos a explicar las distintas opciones de escalado tratadas anteriormente aplicadas a AWS.

3.4.1.1 Escalado vertical

3.4.1.1.1 Esquema básico

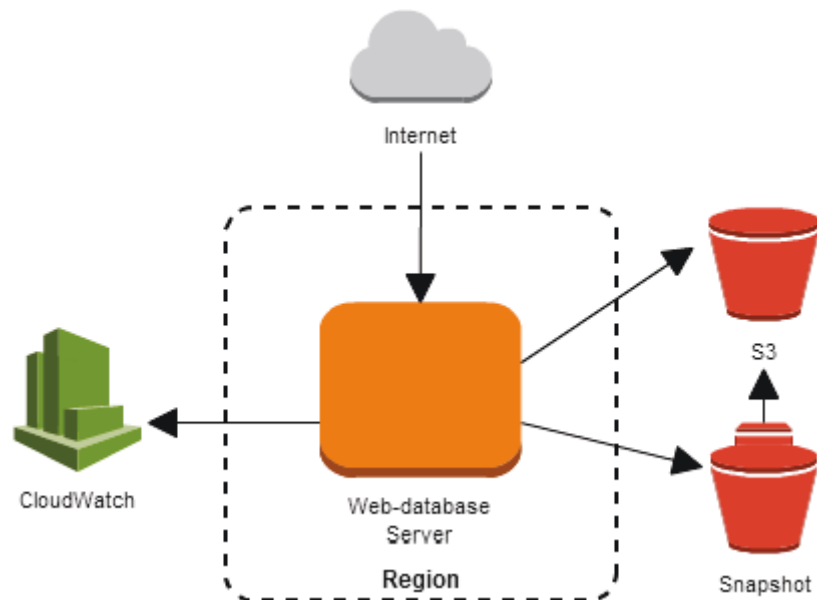


Ilustración 12: Diagrama simple Servidor Web y BD

El esquema anterior (Ilustración 12) es un diseño simple en el que un solo servidor concentra todos los servicios necesarios para poder servir páginas web dinámicas, principalmente servidor web y servidor MySQL.

Además disponemos de un sistema de monitorización del servidor (CloudWatch) y un sistema para almacenar copias /imágenes del mismo y otro para backups.

En este caso, ante un fuerte de demanda que supere la capacidad del servidor no tenemos forma de actuar, por lo que podríamos llegar a quedarnos sin la capacidad de responder peticiones. En caso de ser un aumento paulatino, la única forma de escalar es haciéndolo de forma vertical, es decir, aumentar los recursos de nuestro servidor. Ello obligaría a apagar el servidor, cambiar el tipo de instancia EC2 por una con mayores recursos y volver a iniciarlo, interrumpiendo el servicio durante varios minutos.

3.4.1.1.2 Esquema básico en VPC

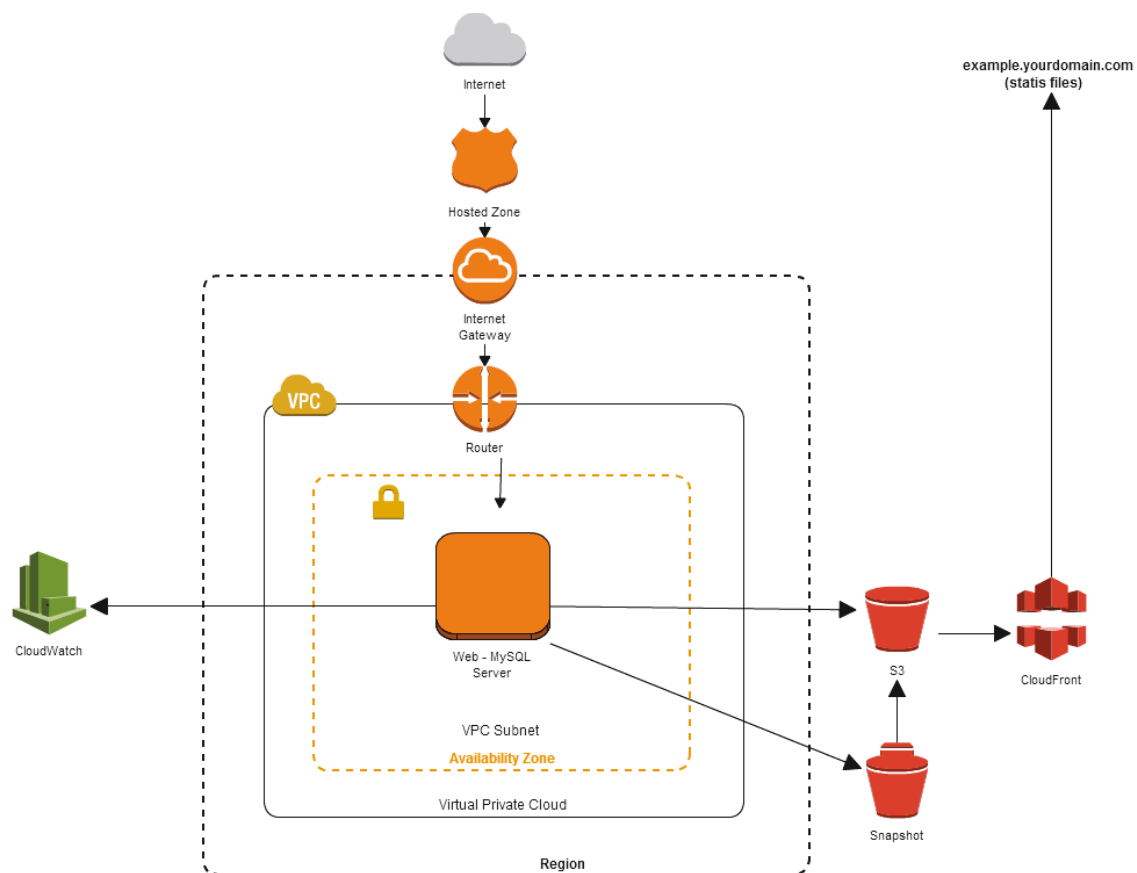


Ilustración 13: Diagrama Servidor Web y BD simple en VPC

El diagrama de la Ilustración 13 presenta el mismo caso que el anterior, pero haciendo uso del servicio de VPC (Virtual Private Cloud) de Amazon.

Una instancia normal de un servidor de EC2 tiene una ip privada y, en caso de que se quiera que sea accesible desde internet, se le asocia una ip pública. Si se sigue la configuración por defecto, la ip privada que se nos asigna está dentro de una subred que, en principio, está compartida con otros usuarios. Por tanto, con la finalidad de tener un entorno lo más privado posible se crea un “Cloud privado” con una subred que conectamos a internet mediante un router y una puerta de enlace.

Debemos empezar a pensar en un sistema más descentralizado en el que se pueda ampliar de forma individualizada los recursos ya que, hasta el momento, lo único que podemos hacer es prever el peor escenario posible y contratar un servidor con recursos suficientes, lo cual repercute directamente en la factura al estar siempre pagando por un servidor que la mayor parte del tiempo está infrautilizando sus recursos.

3.4.1.1.3 Esquema servidor Web y Base de datos en diferentes nodos

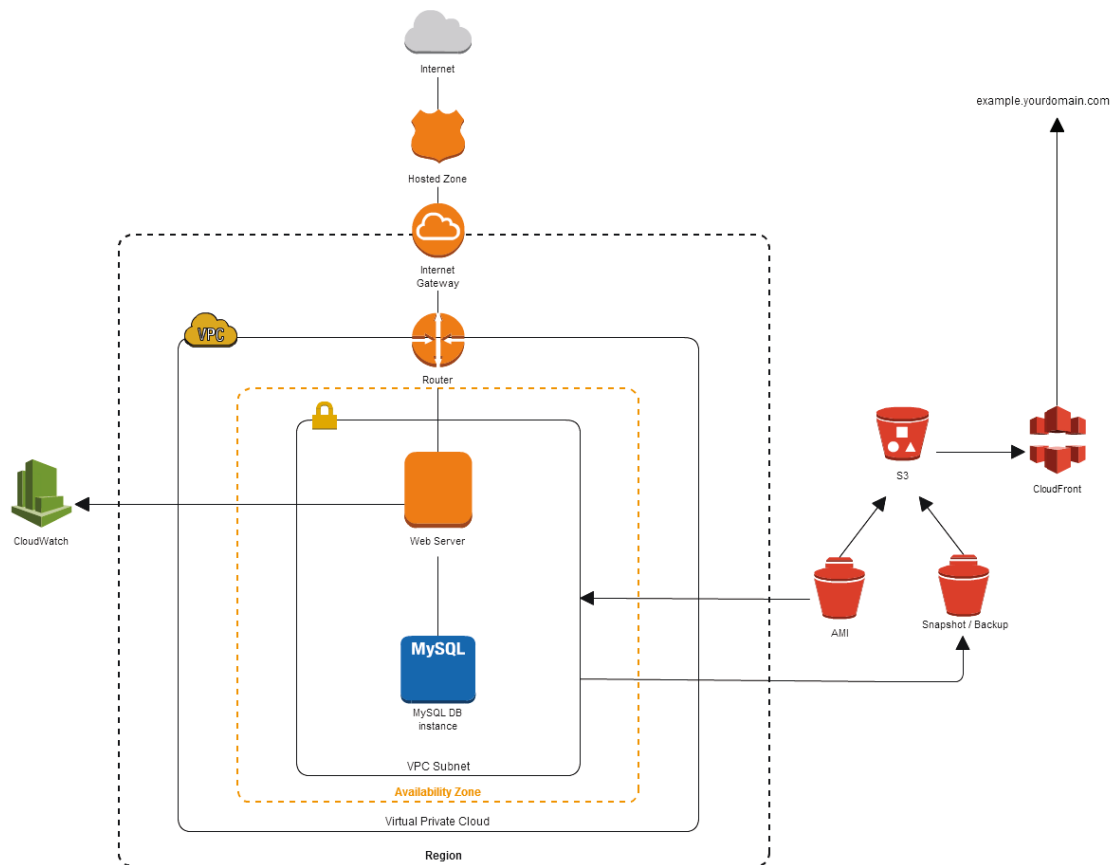


Ilustración 14: Diagrama Servidor Web y BD en diferentes nodos

En la Ilustración 14 pasamos a un sistema más descentralizado donde por un lado tenemos el servidor web y, por otro, el servidor de bases de datos. Esto permite que cada uno de ellos se puedan configurar unos recursos hardware y software acordes a sus necesidades concretas, aumentando el rendimiento de cada sistema además del ya ganado al haber separado los dos sistemas principales.

En este tipo de configuraciones el escalado del servidor web sigue siendo necesariamente vertical, pero en lo que se refiere al servidor MySQL, podemos reescalar horizontalmente, es decir, crear una réplica de lectura [84] del servidor (servidor esclavo) y destinarla a las consultas de lectura y backups, descargando de esta forma, el servidor principal (servidor maestro).

3.4.1.2 Escalado horizontal

3.4.1.2.1 Esquema escalado horizontal Servidor Web y Servidor de Base de datos

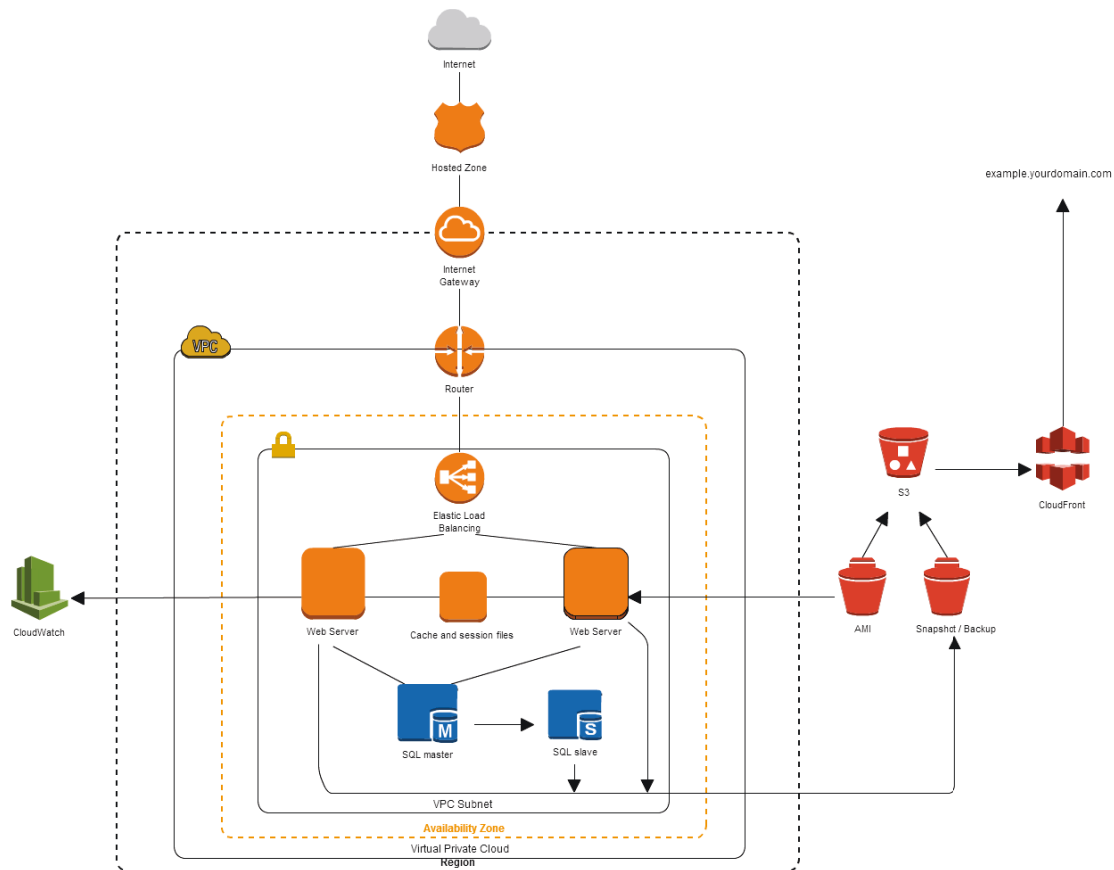


Ilustración 15: Diagrama escalado horizontal Servidor Web y Servidor de Base de datos

En el esquema de la Ilustración 15 pasamos a un entorno en el que hemos escalado de forma horizontal. Partiendo del esquema de la ilustración 14, en la cual ya teníamos separados los dos principales servicios (servidor web y servidor de base de datos), tenemos dos instancias de servidor web idénticas y, mediante un balanceador de carga, permitimos que sean accedidas desde internet. Esto reparte la carga de trabajo web entre dos servidores, teniendo cada uno una copia del código que deben ejecutar.

Con respecto a la base de datos, en este caso tenemos una configuración Maestro / Esclavo, mediante la cual toda inserción o modificación realizada en la base de datos maestra es replicada de forma atómica a la copia esclava. Esto nos permite tener una copia en tiempo real de nuestra base de datos y poder derivar las consultas que sólo impliquen lectura a la instancia esclava, lo que reduce la carga en la base de datos principal. Además, como se puede ver en el diagrama, los backups se realizan sobre la base de datos copia, evitando así sobrecargar la principal.

Con este esquema, si se llegara a necesitar mayor capacidad Web, sólo tendríamos que generar otra instancia idéntica y habríamos escalado nuestro sistema rápidamente y sin ningún tipo de

interrupción del servicio. En el caso de la base de datos, no nos quedaría más opción que plantear un escalado vertical de la base de datos maestra o esclava (en el caso de que fuera ésta la que sufriera saturación). En este segundo caso, otra opción, aunque más complicada, pasaría por la creación de otra réplica de lectura y que ambas fuesen atacadas a través de un balanceador.

En este tipo de arquitectura nos surge la problemática tratada anteriormente: qué hacer con archivos que deben ser accedidos de igual manera por cada instancia (cookies de sesión, archivos de caché e imágenes).

Los archivos de sesión, caché e imágenes deben ser únicos y compartidos por todas las instancias ya que si no, dependiendo de la instancia a la que el cliente sea redirigido, podría tener o no su sesión (login, carro de la compra, etc) del anterior acceso. Normalmente los balanceadores intentan redirigir siempre la misma dirección IP la misma instancia, pero no siempre es posible.

En el caso de las imágenes (contenidos media en general) , es necesario que su ubicación sea compartida debido a que si no para cada producto nuevo o imagen editada podría haber una instancia con imágenes y otra a la que le faltaran.

La forma de solventar este inconveniente es tener un servidor NFS (Network File System) en red con tres ubicaciones compartidas correspondientes a la caché, el directorio de las imágenes y el directorio de las sesiones. En la AMI principal de Magento, montaríamos los respectivos directorios en red apuntando a su correspondiente carpeta en el servidor NFS. De esta forma, si la instancia A genera un nuevo archivo de sesión por un nuevo visitante, todas las demás instancias tendrán acceso a dicha sesión en caso de necesitarlo.

3.4.1.2.2 Esquema con Autoescalado y RDS

Habiendo resuelto el problema del escalado de nuestro sistema, podemos dar un paso más y hacerlo autónomo, de forma que no necesite una continua supervisión del nivel de carga para establecer si se deben añadir o eliminar recursos. Podemos programar un escalado automático de recursos para que, automáticamente, en función de la carga de nuestro servidor se asignen más o menos recursos que permitan absorber picos de carga.

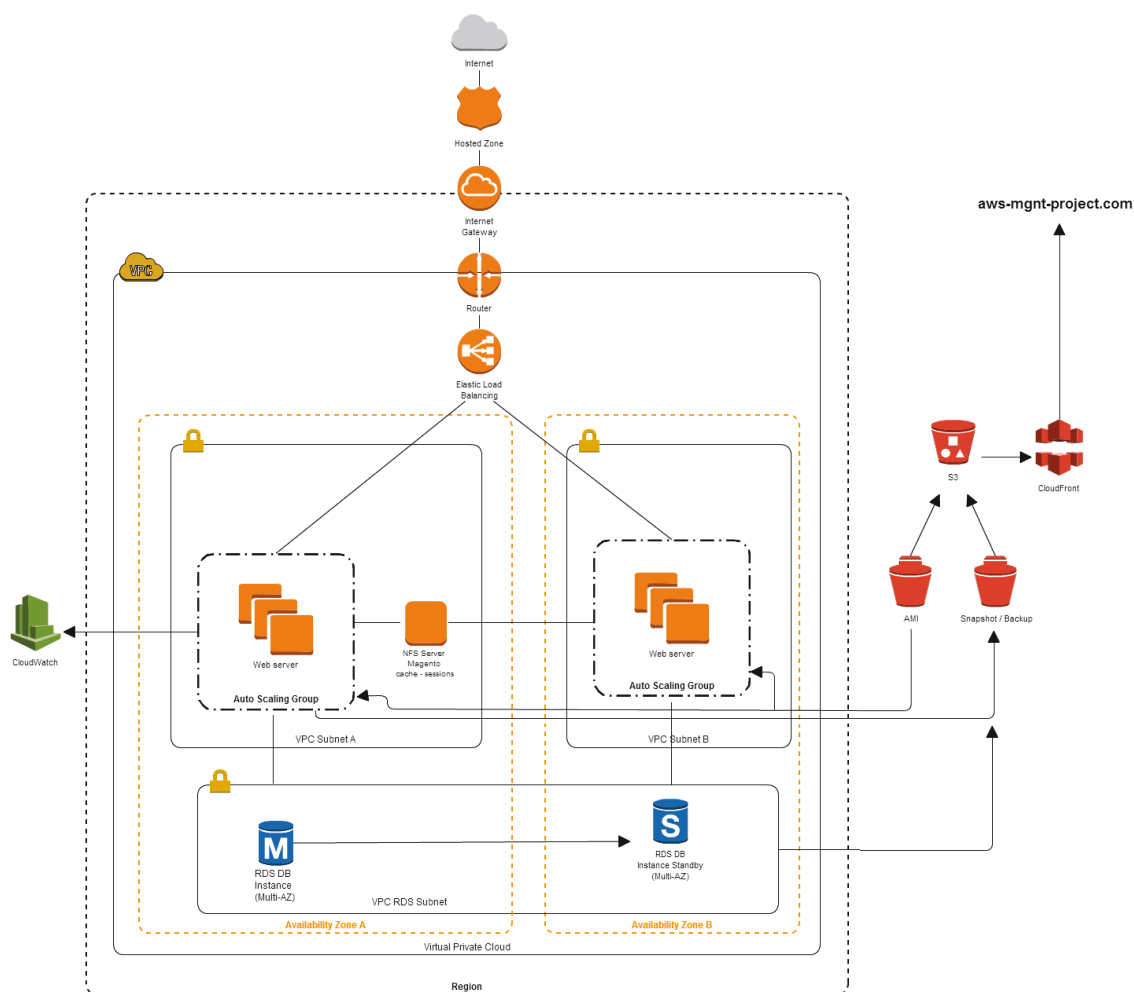


Ilustración 16: Diagrama Sistema Autoescalable y tolerante a fallos en AWS

En el esquema presentado en la Ilustración 16 podemos ver un entorno compuesto por dos zonas de disponibilidad, lo que nos permite garantizar aún mayor disponibilidad al estar protegidos ante una caída del servicio en uno de los CPD [86]. En caso de caída de uno de los dos, el balanceador, que de por sí es redundante en cada zona, seguiría redirigiendo las peticiones a los servidores que están activos en la zona disponible.

Además, el sistema web está monitorizado por el sistema de auto-escalado que, por medio de una serie de reglas pre-configuradas, añade o elimina instancias en función de los datos de carga obtenidos a través de CloudWatch. Los cambios de instancias se intentan realizar de forma que las zonas de disponibilidad se queden lo más equitativas posibles.

Un servidor NFS comparte entre todas las instancias los archivos críticos de los que ya hemos hablado en apartados anteriores (archivos de caché, las sesiones de usuario y los archivos de imagen).

La base de datos se ha configurado a través del servicio RDS MultiAZ, que nos gestiona automáticamente el servidor de base de datos principal y una réplica de lectura en otra zona de disponibilidad. De esta forma, si hay una caída de servicio en la zona principal, AWS

automáticamente gestiona el uso de la réplica de lectura como base de datos principal (proceso conocido como failover), asegurando la disponibilidad del sistema.

De la misma manera, en el momento que la zona principal vuelva a estar operativa, se gestiona de forma automática y transparente para el usuario o administrador del sistema, la puesta en funcionamiento de la arquitectura anterior (proceso conocido como failback).

3.4.1.3 Base de datos EC2 vs RDS

En los primeros diagramas se puede ver cómo el sistema gestor de base de datos relacional (SGBDR[89]) era instalado primero dentro de la misma instancia EC2 que el servidor web, y luego, en su propia instancia independiente. En los dos casos, se requiere una instalación y gestión del SGBDR, de forma que se asegure su disponibilidad, rendimiento y respaldo de los datos. Esta opción nos permite elegir entre distintos sistemas compatibles con MySQL (MariaDB [87] o Percona [88]).

Uno de los requisitos del proyecto es mantener una disponibilidad del 99,9%, por lo que debemos asegurar que el SGDBR tiene una configuración adecuada a esta necesidad.

Para ello deberíamos tener, al menos, una réplica de lectura de la base de datos en otra zona de disponibilidad, la cual nos permitiría:

- Descargar la base de datos principal al redirigir las operaciones de lectura a la réplica (Slave).
- En caso de fallo de la base de datos principal (Master) o desconexión de la zona de disponibilidad donde se encuentra la base de datos principal, asignar la réplica de lectura como nuevo servidor master. Este proceso se conoce como failover o conmutación por error.

Toda la configuración de este sistema y el proceso de failover se ha de realizar de forma manual y por personal técnico cualificado, lo que aumentaría los costes dado que necesitaríamos un equipo Sysadmin [90] que estuviese monitorizando este sistema y actuase ante errores de sincronización y, en caso de fallo del master, actuara en el momento realizando las operaciones de failover necesarias.

Desde el momento en el que se opta por tener un nodo específico para el SGBDR, tenemos la opción de Amazon Web Services Relational Database Service. Este sistema instala y gestiona una instancia MySQL (también Oracle o SQL Server, pero en este caso nos interesa sólo MySQL) permitiendo las siguientes operaciones desde el panel de administración de AWS RDS:

- Gestión, optimización y monitorización del SGDBR.
- Creación de backups (snapshots) automáticamente.
- Creación de réplicas de lectura en diferentes zonas y gestión de las mismas.
- Multi-AZ. Creación automática de réplica de lectura y gestión automática de failover y failback ante fallo del máster.

Esto nos permite que el sistema se encargue de la creación y gestión del SGBDR evitándonos el trabajo que requeriría estar monitorizando 24x7 la disponibilidad del sistema y actuando en consecuencia. Además nos libera de los problemas comunes en este tipo de casos, como son la desincronización de las réplicas. También reduce el tiempo que el sistema está caído al realizar el failover automáticamente y no ser necesaria la modificación la configuración de acceso a la base de datos al realizarse las modificaciones DNS pertinentes en las direcciones de los servidores Master y Slave.

3.4.1.4 Diseño final

De todos los distintos diseños del sistema vistos, implantaremos el representado en la Ilustración 16. Dicha arquitectura nos garantiza un sistema con una disponibilidad muy alta (requisito fundamental) pero además nos permite una gran elasticidad a la hora de incrementar recursos si fuese necesario.

Dado que el escalado horizontal y automático del servidor de bases de datos no es posible, dicho incremento de recursos recae sobre los servidores web. De esta forma, se puede calcular un sistema que en condiciones de baja demanda disponga de un número mínimo de instancias, pudiendo ser éstas, además, instancias de menor rendimiento.

A pesar de que seguimos limitados por la capacidad del servidor de bases de datos, una buena configuración de sistemas de caché reduce en gran medida la demanda de peticiones a la base de datos, lo que nos permite un enorme margen de crecimiento de nuestra capacidad de gestión de peticiones simplemente aumentando los recursos web cuando sea necesario.

3.5 Metodología de desarrollo

3.5.1 Herramientas de desarrollo

En la siguiente tabla resumen se exponen las diferentes herramientas usadas para el desarrollo de este proyecto:

| Herramientas de desarrollo | |
|----------------------------|---|
| Virtualbox [91] | Software de virtualización. Permite la creación de máquinas virtuales con el sistema operativo, el software y las herramientas necesarias para poder emular el sistema que nos encontraríamos en producción |
| Vagrant [92] | Herramienta para la creación y configuración de entornos de desarrollo virtualizados. |
| Puphpet [93] | Aplicación web que permite generar una configuración de inicio para Vagrant de forma sencilla y rápida. |
| Git [94] | Software de control de versiones |
| PhpStorm [95] | IDE [96] (Integrated Development Environment) para desarrollo de PHP. Permite la conexión con host remoto para la sincronización de archivos. Control de versiones integrado (Git entre otros). Múltiple módulos, como Magento, para el desarrollo específico de Magento. |
| Phpmyadmin [97] | Herramienta para administrar MySQL a través de una interfaz web. |

Tabla 107: Herramientas de desarrollo

3.5.1.1 Vagrant y Puppuppet

Vagrant es una herramienta que nos permite virtualizar entornos de desarrollo web (pre-programados a través de un fichero de configuración) con un sólo comando: “vagrant up”.

Puppet nos permite crear de forma sencilla el fichero de configuración que usará Vagrant. Podemos elegir el sistema operativo, paquetes del mismo a instalar, configuración del servidor web, lenguajes de programación, configuración de la base de datos y un largo etcétera.

3.5.2 Entornos de desarrollo

A la hora de comenzar el desarrollo de un proyecto web, se deben definir al menos dos entornos diferentes de trabajo, aunque lo ideal son tres:

- Desarrollo
- Preproducción
- Producción

3.5.2.1 Desarrollo

Entorno normalmente local en el que se realiza la mayor parte del trabajo del proyecto. Herramientas de control de versiones como Git permiten que varios desarrolladores puedan estar trabajando a la vez en el mismo proyecto.

A pesar de que se trabaje en local, podemos emular un entorno de producción gracias a Vagrant. Por ejemplo, si nuestro futuro proyecto estará alojado en un servidor en AWS en una imagen de Debian, con un servidor web apache y módulos de PHP específicos, podremos recrear ese sistema y trabajar de forma que no haya diferencia alguna (en lo que a configuración software se refiere) con el servidor de producción. Además, nos ayudará a que el paso de preproducción a producción requiera menos tiempo, al ya haber de por sí pocas diferencias con respecto a desarrollo.

3.5.2.2 Preproducción

Entorno exacto a producción, tanto en lo relativo al software desarrollado como al contenedor/servidor donde se ejecuta. El objetivo es recrear exactamente el entorno de producción, muchas veces muy diferente al usado para el desarrollo del proyecto, y así poder realizar todas las pruebas necesarias para certificar el correcto funcionamiento de todo nuestro sistema.

Si hemos hecho uso de Vagrant en el entorno de desarrollo, para este paso podemos realizar, si cabe, una prueba más real haciendo uso del mismo fichero de configuración que creamos para generar el entorno de desarrollo pero, en este caso, para crear ese mismo sistema en el proveedor de servidores que usemos (si éste está soportado por Vagrant).

De esta forma, en lugar de generarse un entorno a medida para nuestro sistema de forma local haciendo uso de VirtualBox, lo podemos crear directamente en una instancia EC2 de AWS, lo cual, dejará ya pocas probabilidades de fallo una vez pasemos a producción.

3.5.2.3 Producción

Paso final. El software que hemos desarrollado está siendo ejecutado en su ubicación final pública. Si hemos usado de forma correcta los dos anteriores entornos, no deberían surgir problemas.

3.6 LOPD

La Ley Orgánica de Protección de Datos de Carácter Personal [98] es una Ley Orgánica española que tiene por objeto garantizar y proteger, en lo que concierne al tratamiento de los datos personales, las libertades públicas y los derechos fundamentales de las personas físicas, y especialmente de su honor, intimidad y privacidad personal y familiar.

Cualquier empresa española está obligada a cumplir con ella, por lo que tanto nosotros debemos hacerlo como nuestros proveedores. Ello significa que si contratásemos un proveedor de servicios cloud español y con sus sistemas ubicados en territorio nacional, nuestros datos, en lo referente a su almacenamiento y tratamiento en los sistemas de dicha empresa, estarían cumpliendo con la ley.

Si por el contrario contratásemos los servicios de un proveedor extranjero, nosotros seguimos obligados a cumplir con la ley pero ellos no (o al menos no de la misma manera). Las leyes europeas son equiparables a las españolas, pero debemos estar seguros de cumplirlas. No es lo mismo que nuestro proveedor tenga ubicados sus sistemas en la Unión Europea que fuera de ella.

Para estar seguros de que un proveedor extranjero cumple con los requisitos mínimos de la Directiva sobre protección de datos de la Unión Europea, éste debería haber suscrito las disposiciones de privacidad Safe-Harbor [99], aunque es cierto que dicha disposición ha sido declarada nula recientemente por el Tribunal de Justicia de la Unión Europea basándose en la revelaciones hechas por Edward Snowden.

El proveedor que hemos elegido, Amazon Web Services, además de permitir la elección de Irlanda como ubicación de los sistemas, recientemente ha visto aprobado por la UE su Acuerdo de Procesamiento de Datos (APD por sus siglas en inglés). Enlaces [1](#) y [2](#).

Por lo tanto, independientemente del centro de datos de AWS que usemos, se considerará que cumplen con las leyes europeas de privacidad.

En cualquier caso, es imprescindible dar de [alta](#) nuestro sistema en la Agencia Española de Protección de Datos.

3.6.1 LOPD en nuestro proyecto

A pesar de los recientes acuerdos, para la elaboración de nuestro proyecto se ha decidido ir más allá e intentar proteger los datos al máximo.

Para ello, se ha implementado un entorno VPC para nuestro sistema, es decir, una nube privada a la que nadie más que nuestras máquinas (con su rango de IP privado) tienen acceso. La única forma de acceder es través de un balanceador web a través de los puertos estándar 80 y 443, que internamente redirige las peticiones a los servidores web. No se tiene acceso directamente desde el exterior ni a los nodos web, ni a los nodos de bases de datos ni a ningún otro nodo.

En los momentos en los que sea necesario, la comunicación entre el cliente de la tienda y nuestro sistema se realiza mediante comunicación cifrada SSL.

El almacenamiento de la base de datos está cifrado.

En los nodos web no se almacena información sensible de los clientes, por lo que se decide no cifrarlos debido al coste que implicaría, ya que el tipo de instancia elegido no admite cifrado de disco y habría que contratar una instancia mayor.

Nuestro sistema no almacena datos de pago ni gestiona los mismos, sino que provee las redirecciones necesarias a las pasarelas de pago de entidades bancarias.

4. Implementación / Resultado

4.1 Requisitos del proyecto

Finalmente, el software elegido para implementar y como gestión básica de la empresa, ha sido Magento por las razones expuestas en puntos anteriores.

Como proveedor nos hemos decantado por Amazon Web Services, implementado el esquema de la Ilustración 16, lo cual nos proporciona un sistema con las siguientes características:

- **Privado:** Se ha creado una VPC (Virtual Private Network) propia de forma que tengamos un sistema aislado del resto dentro de la infraestructura de AWS.
- **Autoescalable:** En el apartado de los servidores web, se dispone de un servicio de autoescalado que, partiendo de un mínimo de dos servidores web, en función de la carga irá levantando nuevas instancias o eliminándolas. En el apartado de la base de datos, se ha reforzado la capacidad del sistema dotándolo de una réplica de lectura, de forma que las consultas de sólo lectura y los backups se realicen sobre la misma, descargando el nodo principal. Si fuera necesaria más capacidad, podríamos optar por la creación de más réplicas de lectura o, realizar escalado vertical sobre el nodo principal (y la/s réplica/s de lectura si fuese necesario).
- **Alta disponibilidad:** Para cumplir con los requisitos de disponibilidad, se han creado dos zonas de disponibilidad “Zona A” y “Zona B”, existiendo, dentro de cada una de ellas, una subred (“Subred A” y “Subred B” respectivamente) de nuestra VPC de forma que pertenezca a nuestra VPC. Dentro de cada subred se encuentra el sistema de autoescalado comentado en el punto anterior. De esta forma, aseguramos la disponibilidad a la vez que tenemos una gran capacidad de escalado. El sistema de autoescalado también realiza funciones de vigilancia del estado de las instancias, sustituyendo aquellas que no estén funcionando correctamente. En el apartado de bases de datos, hacemos uso del Servicio RDS de AWS con la funcionalidad Multi-AZ. Esto nos permite una gestión de la base de datos del sistema y una alta disponibilidad, ya que el complemento Multi-AZ despliega una réplica de lectura de la base de datos en otra zona distinta a la principal y, en caso de caída del principal nodo, realiza un failover sobre a réplica automático (y el failback posterior).
- **Integrado:** Todas las funcionalidades que se requieren para el correcto funcionamiento del sistema, así como su escalado de futuro, están cubiertas por los servicios de AWS, por lo que no necesitaríamos tener varios proveedores, unificamos los gastos, se reducen costes y tenemos para todos ellos un modelo de pago por uso (IaaS). El envío de correo electrónico lo haremos haciendo uso del servicio Simple Email Service (SES) [100] de AWS

- **Bajo coste:** Al implementar todos los servicios necesarios en un solo proveedor y seguir éste un modelo de pago por uso, obtenemos una reducción de coste significativa si lo comparamos con otros servicios de la competencia.

| Ventajas | Desventajas |
|--|---|
| <ul style="list-style-type: none"> ✓ Capacidad mundial de ventas sin necesidad de despliegue internacional de la empresa. ✓ Control absoluto de la venta de los productos y su envío desde un solo sistema. ✓ Escalabilidad del sistema en función de la demanda con capacidad de reacción en minutos. ✓ Alta protección ante caídas o pérdida del servicio. ✓ No requiere la adquisición de una gran infraestructura hardware ni de los costes que ello conlleva. ✓ No es necesario un numeroso departamento TI. Incluso se puede externalizar la gestión de AWS. | <ul style="list-style-type: none"> ✗ Se necesita personal técnico especializado en la gestión de AWS. ✗ Posibilidad de altos costes en caso de incorrecta configuración o bajo control del sistema. ✗ A pesar de ser Magento un software con una gran comunidad y grandes perspectivas de futuro, en caso de dejar de recibir soporte sería un proyecto condenado a morir al ser difícil que una sola empresa pudiera acometer los costes de mejora y reparación de vulnerabilidades que fueran apareciendo. |

Tabla 108: Ventajas – Desventajas Magento AWS

4.2 Implementación de la arquitectura diseñada

Procederemos a implantar la arquitectura diseñada en la Ilustración 16, en la cual encontramos un servidor web por zona de disponibilidad, un servidor NFS que almacena las sesiones de usuario, imágenes y caché y un servidor de base de datos.

4.2.1 Región

AWS permite el despliegue de sus servicios en diferentes regiones del mundo, lo que permite elegir la región que más cerca de los potenciales clientes o que por su ubicación cumpla determinados requisitos legales, como por ejemplo, protección de datos.

Hay que tener en cuenta que no todas las regiones disponen de todos los servicios de AWS.

En nuestro caso hemos elegido el centro ubicado en Irlanda. De esta forma alojamos nuestros datos dentro de la Comunidad Europea.

4.2.2 Zonas de disponibilidad

Las regiones de AWS disponen de varias zonas de disponibilidad diferentes que no son más que centros de datos alejados entre sí pero interconectados por una red de baja latencia. De esta forma, un fallo en una ubicación no se transmite al resto.

Esto nos permite tener dentro de la misma región un sistema redundante de forma que, aunque una zona de disponibilidad sufriera algún tipo de percance, nuestro sistema pudiera seguir dando servicio a nuestros clientes.

4.2.3 Entorno VPC

Un entorno VPC (Virtual Private Cloud) o Cloud Privado Virtual nos permite crear nuestra propia nube privada dentro de la infraestructura de AWS. De esta forma obtendremos un entorno aislado dentro de la red interna de nuestra región donde podremos disponer de todos los servicios de AWS.

AWS por defecto no permite la entrada de ningún tipo de conexión a nuestras instancias a no ser que dicha conexión haya sido permitida en el Grupo de Seguridad asociado a la instancia en cuestión. Crear una VPC nos permite aislar nuestro sistema de la red interna de AWS, de forma que tengamos nuestro propio “Cloud” presente en las zonas de disponibilidad que consideremos necesarias y las comunicaciones entre nuestros nodos sean privadas incluso dentro de la propia red de AWS.

El servicio VPC de AWS permite diferentes configuraciones por defecto:

- VPC con subredes públicas: Permite crear un entorno en el que las subredes tengan acceso a internet.
- VPC con subredes públicas y privadas: Permite crear un entorno en el que existan subredes con acceso a internet y subredes privadas con acceso al exterior a través de una subred pública pero sin acceso directo desde internet.
- VPC con subredes públicas y privadas y acceso VPN: Mismo caso que el anterior, pero además se habilita una conexión VPN (Virtual Private Access) [101] al VPC.
- VPC con subred privada y acceso VPN: Se habilita una subred privada únicamente accesible a través de una conexión VPN.

El servicio de VPC de AWS no tiene coste, por lo que, si fuese necesario, podríamos crear varias VPCs sin coste adicional.

4.2.3.1 Creación de la VPC

Para el desarrollo de nuestro proyecto es suficiente con la creación de subredes públicas. Mediante los Grupos de Seguridad, controlaremos los accesos desde el exterior. Más adelante se crearán subredes privadas, pero serán los propios servicios de AWS quienes lo hagan de forma automática como parte de la configuración de los mismos.

Para que la VPC tenga acceso al exterior, es necesaria la configuración de una Puerta de Enlace a Internet o Internet Gateway además de unas tablas de enrutamiento o Route Tables que se encarguen dirigir los datos de forma local entre las distintas subredes o hacia el exterior.

Durante la configuración del VPC, deberemos seleccionar el Bloque CIDR (Classless Inter-Domain Routing) [102] de nuestra red, es decir, el rango de red que usaremos, el cual definirá el número de máquinas que podremos crear. De igual forma deberemos hacerlo para las subredes que necesitemos.

Hemos configurado un Bloque CIDR: 10.0.0.0/24, el cual nos permite disponer de hasta 251 hosts. Teniendo en cuenta que se han previsto dos zonas de disponibilidad (por tanto dos subredes), 125 host para cada zona de disponibilidad es una cantidad razonable de ips disponibles que, en caso de incremento de accesos al sistema, nos permitiría escalar nuestros recursos.

De esta forma, la configuración de nuestro Cloud Privado se quedaría de la siguiente forma:

- Una VPC “Mgnt-project-VPC” creada en la región “eu-west” con un bloque CIDR 10.0.0.0/24
- Una subred “Mgnt-project-Subnet-1a” asignada a la zona de disponibilidad “eu-west-1a” con un bloque CIDR 10.0.0.0/25
- Una subred “Mgnt-project-Subnet-1b” con un bloque CIDR 10.0.0.128/25

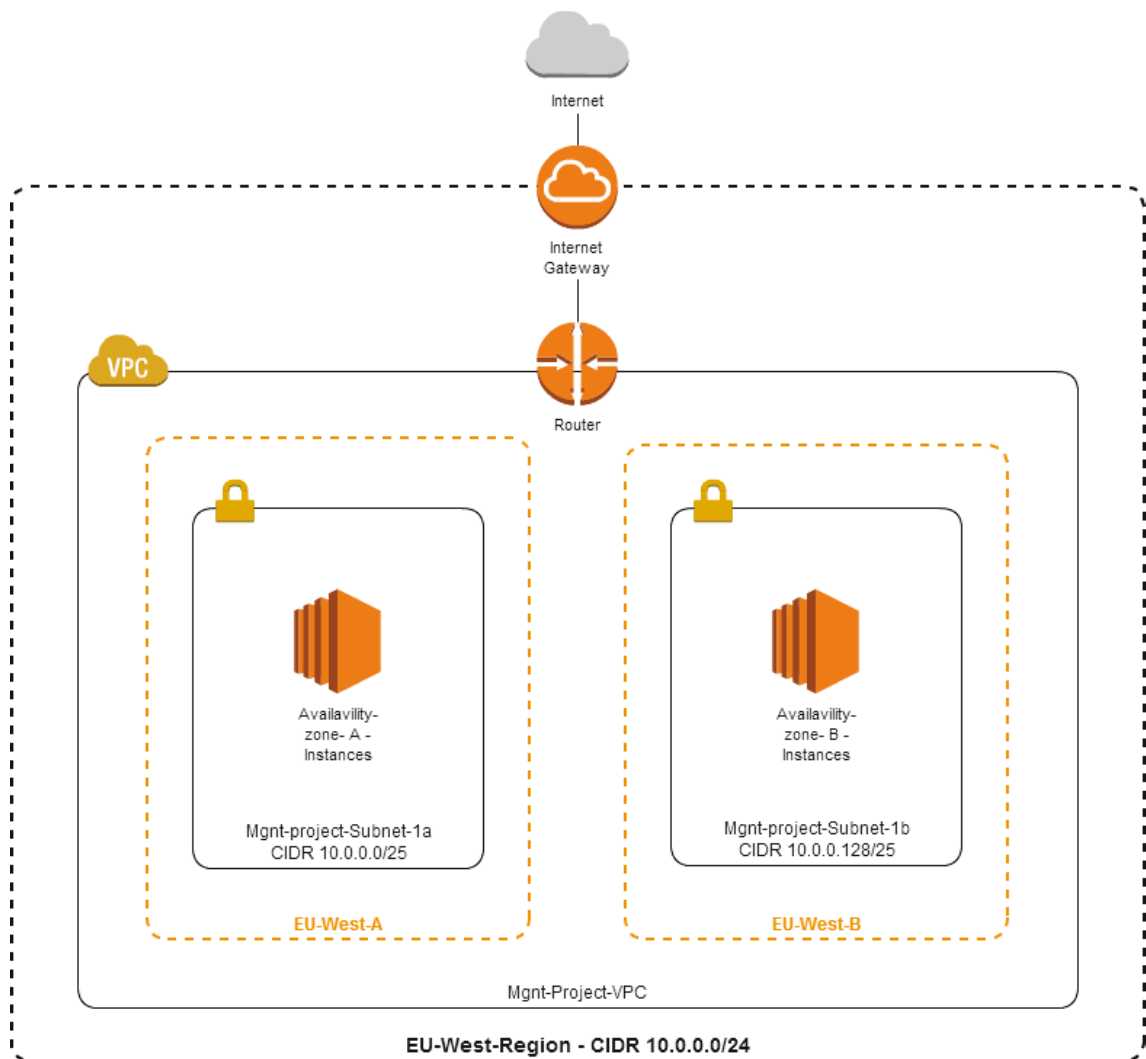


Ilustración 17: Diagrama VPC

4.2.4 Instancia RDS (Base de datos)

Las instancias RDS nos permiten la gestión de las bases de datos de forma sencilla, con una alta disponibilidad y una gran automatización.

Para nuestro proyecto las configuraremos con el motor MySQL y Multi-AZ en una instancia m3.medium con las siguientes características:

- 1 x vCPU
- 3,75GB RAM
- 20GB SSD de propósito general (60 IOPS) cifrado

Con el fin de cumplir al máximo con la normativa de la LOPD (Ley Orgánica de Protección de Datos) y a pesar de que nuestro sistema se desarrolla en un Cloud privado no accesible por terceros, todos los sistemas de almacenamiento usados (y sus respectivos snapshots o backups) serán cifrados haciendo uso del servicio que AWS provee para ello. Todos los datos son cifrados usando una clave única de 256 bits gestionada por el sistema de credenciales de AWS.

El servicio RDS usa su propia subred, por lo que para proceder a la configuración del servicio, lo primero debemos crear una subred que pertenezca a nuestra VPC y que esté disponible en las dos zonas de disponibilidad que conforman la VPC creada en pasos anteriores.

De esta forma, podremos activar el servicio Multi-AZ, el cual gestiona de forma automática y transparente la creación de la instancia principal y su réplica en una zona de disponibilidad diferente. En caso de ser necesario, realiza un proceso de failover automático, esto es, la réplica es conectada asignándose como base de datos principal. Así, aunque nuestro sistema siga teniendo configurada la base de datos principal, las peticiones son automáticamente redirigidas (haciendo los cambios oportunos en el dns interno) hacia la base de datos sea réplica, ahora convertida en principal o “Máster”.

Hemos de tener en cuenta que la réplica generada por el sistema MultiAZ no es accesible como réplica de lectura, sino que es una copia gestionada por RDS para ser utilizada en caso de fallo o desconexión de la principal.

Para la configuración deberemos configurar los siguientes datos:

| RDS | |
|----------------------------|----------------------------------|
| Parámetro | Valor |
| Motor de base de datos | MySQL |
| Tipo de Instancia | db.m3.medium |
| Tipo de almacenamiento | SSD de propósito general |
| Tamaño de almacenamiento | 20GB |
| Identificador de instancia | /* Identificador */ |
| Nombre de la base de datos | /* Nombre de la base de datos */ |
| Usuario | /* Usuario */ |
| Contraseña de usuario | /* Contraseña */ |
| VPC | /* Nuestra VPC */ |

| | |
|--|--|
| Subred | /* Subred para RDS */ |
| Zona de disponibilidad preferida para Master | /* Zona de disponibilidad elegida */ |
| Puerto de acceso a la base de datos | 3306 |
| Grupo de Seguridad | /* Grupo de seguridad específico para RDS */ |
| Habilitar cifrado de disco | Sí |
| Configuración Backups | Sí |
| Mantenimiento | Sí |

Tabla 109: Parámetros RDS

Se generan backups (snapshots) automáticos diarios, pudiéndose elegir el tiempo que serán guardados por AWS. Los backups siempre se realizan sobre la instancia réplica para no afectar a al rendimiento de la instancia principal.

AWS realizará un mantenimiento periódico de nuestras instancias RDS, de forma que siempre están actualizadas.

Una vez hecho todo esto, tenemos nuestra base de datos en funcionamiento y correctamente asegurada tanto su disponibilidad como sus datos.

4.2.5 EC2

El servicio EC2 de AWS nos permite aprovisionar máquinas virtuales en función de nuestras necesidades. Podemos elegir entre varios tipos de instancias, cada uno de ellos con unas características hardware diferentes, como número de núcleos virtuales, memoria RAM, o el ancho de banda disponible.

Las instancias EC2 (Elastic Cloud Computing) son creadas en base a una imagen AMI (Amazon Machine Image). Dichas imágenes pueden proceder de:

- AWS Marketplace: Son creadas bien por AWS o por empresas. En muchos casos las imágenes viene ya pre-configuradas con un sistema y funcionalidad concreta (Ej.: LAMP [103]). En algunos casos se cobra por su uso.
- Community: Creadas por usuarios y publicadas de forma que cualquiera puede hacer uso de ellas.
- Privadas: Creadas por uno mismo a partir de una imagen o Snapshot del volumen de una instancia. Si se hacen públicas, pasarían a ser parte de “Community”.

En nuestro caso partiremos de una imagen pública de la distribución Linux Debian, la cual modificaremos instalando las dependencias necesarias para la función que deban desempeñar.

Como último paso antes de la creación de una instancia es conveniente generar un par de claves para el acceso seguro a nuestras instancias. Dichas claves se componen de una clave pública y otra privada que nos permitirán la conexión a la instancia a través de SSH [104] sin necesidad de especificar una contraseña.

La clave pública se queda registrada en AWS y es insertada en las instancias en el momento de su creación. La clave privada la deberemos conservar nosotros de forma segura y es la que debemos usar en nuestro cliente SSH.

Dependiendo de la imagen de Linux que usemos puede o no estar deshabilitado el acceso root o con contraseña al uso, por lo que la forma más sencilla y segura de acceder a nuestras instancias de forma directa (en caso de ser necesario) es haciendo uso de este sistema de claves.

4.2.5.1 Instancias servidor Web

Para la creación/ configuración de la instancia web nos basaremos en una imagen de Debian 8 (Jessie) que instalaremos en una instancia de tipo t2.medium, la cual incorpora el siguiente hardware:

- 2 x vCore
- 4GB RAM
- 8GB SSD de propósito general (24 IOPS)

Las instancias t2 de Amazon [105] proporcionan un rendimiento de CPU por ráfagas, es decir, están creadas pensando en un uso discontinuo de la CPU. Se tiene un rendimiento base por núcleo de alrededor del 40% de un núcleo normal de una instancia de rendimiento fijo (m4, r3, etc) y se dispone de una serie de créditos que aumentan en los momentos en los que el uso de CPU está por debajo del 40% y disminuyen cuando lo sobrepasa. Esto permite obtener CPU de alto rendimiento pero con un uso discontinuo. Su uso está indicado para servidores web siempre y cuando no tengan de manera constante un uso excesivo de CPU. En determinados momentos en los que tengamos un pico de consumo de CPU, con este sistema podemos obtener, de forma temporal, un mayor rendimiento de CPU que con instancias de otros tipos con rendimiento fijo.

En el caso del servidor web no usaremos un volumen de almacenamiento cifrado por varios motivos:

- AWS no permite por defecto el cifrado de los disco de inicio (boot flag), por lo que disponer del mismo requeriría:
 - Añadir un segundo disco cifrado
 - Un tipo de instancia igual o superior a t2.large. Las instancias inferiores no permiten almacenamiento cifrado.

- En las instancias Web no se guardan contenidos sensibles (Ej: datos de clientes).

Comenzamos con la instalación del software base necesario:

- Servidor web Apache [106].
- Intérprete de PHP [107].
- Dependencias para la conexión entre PHP, Apache y MySQL
- Agregaremos memoria de intercambio o Swap.

Para ello se utilizará el gestor de paquetes APT, haciendo uso del comando apt-get.

4.2.5.1.1 Instalación PHP

Instalación de php:

```
apt-get install php5 php5-mhash php5-mcrypt php5-curl php5-cli php5-mysql php5-gd
```

De esta forma hemos instalado el intérprete de PHP y todos los módulos necesarios para que la comunicación entre Apache, PHP y MySql sea posible. Además, se han tenido en cuenta las dependencias de Magento:

- php5-mhash -> Es necesaria para crear sumas de comprobación, resúmenes de mensajes, códigos de autenticación de mensajes
- php5-mcrypt -> Proporciona algoritmos de bloques tales como DES, TRipleDES, Blowfish y más.
- php5-curl -> permite conectarse y comunicarse con diferentes tipos de servidores y diferentes tipos de protocolos como http, https, ftp y más.
- php5-gd -> Necesario para crear y manipular ficheros de imágenes en formatos de imagen como GIF, PNG, JPEG y más.

4.2.5.1.2 Instalación Apache

Instalación básica:

```
apt-get install apache2
```

Apache tiene disponibles varios módulos [108] que modifican la forma en la que gestiona las peticiones web que recibe:

- Multiprocesamiento tipo Prefork (mpm-prefork).
- Multiprocesamiento tipo Worker (mpm-worker).
- Multiprocesamiento tipo Event (mpm-event).
- Multiprocesamiento tipo ITK (mpm-itk todavía en fase experimental)

De todos ellos, los dos primeros son los más comunes.

Multiprocesamiento tipo Prefork (mpm-prefork) [109]

Método de multiprocesamiento por defecto. Crea diferentes procesos independientes para manejar las diferentes peticiones. Esta técnica de crear varios procesos se la denomina forking, de ahí el nombre mpm-prefork. Al iniciar Apache Prefork se crean varios procesos hijo (el número varía en función de nuestra configuración).

Ofrece una mayor compatibilidad con diferentes módulos por no hacer uso de Threads (hilos).

Multiprocesamiento tipo Worker (mpm-worker) [110]

Hace uso de procesos e hilos al mismo tiempo, es decir, combina las técnicas de forking y threading. Al iniciar Apache Worker se crean varios procesos hijo y a su vez cada proceso hijo emplea varios threads. Con esto se consigue que cada proceso hijo pueda manejar varias peticiones simultaneas gracias a los threads. Emplea menos recursos de CPU y RAM al requerir de un menor número de procesos hijo.

Requiere de módulos que soporten trabajar con threads, por lo que no se puede usar como manejador (handler) de php el módulo que se instala por defecto (mod_php) si no que se ha de usar php-fpm, que sí soporta threads.

| Manejadores de PHP | | |
|--------------------|--------------------------------|-------------------------|
| | Mod_php | Php-fpm |
| Ventajas | Velocidad. Fácil configuración | Mejor uso de recursos |
| Inconvenientes | Mayor consumo de recursos | Configuración más ardua |

Tabla 110: Tabla resumen características manejadores PHP

Por lo tanto, se configurará apache para que funcione con Multiprocesamiento tipo Worker. Para ello deshabilitaremos el módulo original y habilitaremos mpm-worker. Además, se deshabilitarán los módulos no compatibles con threads y los sustituiremos:

```
apt-get install apache2-mpm-worker libapache2-mod-fastcgi php5-fpm
a2dismod php5 mpm_prefork
a2enmod mpm_worker actions fastcgi alias
```

Se configuraría php-fpm y, después de reiniciar apache la instalación habría finalizado. Una vez se realicen pruebas de carga, se puede ajustar la configuración de apache /etc/apache2/apache2.conf de forma que, por ejemplo, se modifique el número mínimo de servidores apache corriendo de forma paralela o el número máximo de cliente por servidor.

4.2.5.1.3 Espacio de intercambio o Swap

Se define como una zona del disco (un fichero o partición) usada para guardar las imágenes de los procesos que no han de mantenerse en memoria física.

Su existencia se debe a la necesidad de vaciar la memoria RAM para que nuevos procesos con mayor prioridad hagan uso de ella. En sistemas con una gran cantidad de memoria RAM no tiene por qué llegar a usarse, es más, su uso ralentiza el servidor ya que cualquier lectura o escritura en disco es increíblemente lenta si es comparada con la velocidad requerida para realizar el mismo trabajo en memoria RAM.

En nuestro caso hemos creído conveniente disponer de 1GB de Swap [111].

Hemos de tener en cuenta que, aunque nuestro sistema aumentará sus recursos cuando se enfrente a un aumento de la demanda, existe una ventana de tiempo de entre 5 y 10 minutos desde que dicho aumento es iniciado hasta que es efectivo. Si el aumento de demanda provoca un colapso de la memoria RAM y no hay un espacio de intercambio donde “descargarla” se entra en un proceso llamado OOM (Out of Memory) [112], el cual, para liberar memoria comienza a matar procesos. Esto puede provocar pérdidas de ventas, de datos y, en el peor de los casos, un colapso del servidor.

En el ejemplo anterior, haciendo uso del espacio de intercambio éste permitirá que el servidor pueda seguir gestionando peticiones mientras se preparan otros servidores para ayudar a gestionar la creciente demanda.

Para habilitar el espacio de intercambio usaremos los siguientes comandos:

```
touch /swapfile
dd if=/dev/zero of=/swapfile bs=1M count=1024
mkswap /swapfile
chmod 0600 /swapfile
swapon /swapfile
```

Para que la partición de Swap sea montada automáticamente al iniciar la máquina, lo indicamos en el fichero fstab:

```
nano /etc/fstab  
/swapfile swap swap defaults 0 0
```

4.2.5.2 Instancia NFS

El servicio NFS (Network File System) posibilita que distintos sistemas conectados a una misma red puedan acceder a ficheros remotos de la misma forma que los harían a ficheros locales.

Al tener un número indeterminado (mínimo dos) de instancias web que ejecutarán Magento, necesitamos un lugar común en el que las distintas instancias alojen los archivos de caché, sesiones de usuarios y las imágenes de los productos, para que todos compartan los mismos datos.

4.2.5.2.1 Single Point Of Failure en NFS

En la arquitectura que finalmente elegimos (Ilustración 16 – Diagrama Sistema Autoescable y tolerante a fallos en AWS) el servidor NFS es un punto de fallo o SPOF (Single Point Of Failure) [113] ya que, en caso de caída del servidor, al no tener el NFS redundancia el sistema se queda bloqueado al perder la conexión al servidor NFS. En cualquier caso, Magento no puede funcionar si no tiene acceso ya no a la caché, si no al directorio media y al directorio de almacenamiento de las sesiones de usuario.

Se ha estudiado la implementación de diversas soluciones de forma que fuera tolerante a fallos, encontrándose en cada uno de ellas diversos problemas.

4.2.5.2.1.1 Autoescalado NFS

Al igual que en las instancias web, se crearía un grupo de autoescalado que controlara que siempre hay una instancia en funcionamiento.

- ✖ **Problemas:** La nueva instancia se crea a partir de una imagen AMI, por lo que para que los datos de caché, sesiones de usuario e imágenes estuvieran actualizados, habría que mantener algún tipo de copia en tiempo real desde la instancia principal para que, en caso de fallo, la nueva instancia pudiese copiar esos datos. Además, una nueva instancia tarda unos minutos en estar operativa. Si le añadimos el tiempo de sincronización desde la copia y el tiempo desde que la instancia empieza a fallar hasta que es detectada, podemos estar hablando de entre 5 y 10 minutos durante los cuales el sistema estaría inoperativo. A ello, habría que sumar el inconveniente del coste adicional que implica el tener operativo el sistema de auto-escalado (ELB).

4.2.5.2.1.2 GlusterFS (Gluster File System)

Permite la redundancia de los datos compartidos en red entre varios nodos.

- ✖ **Problemas:** No recomendado por los desarrolladores para el almacenamiento de archivos caché o sesiones de usuario por motivos de rendimiento. Además, implica un coste fijo por la redundancia de los datos más el coste que implica el tener operativo el sistema de auto-escalado (ELB) [114]

4.2.5.2.1.3 S3FS (S3 File System)

S3FS [115] es un sistema de archivos basado en FUSE [116] que permite montar un bucket de S3 en un sistema local, como si de una carpeta del sistema se tratara.

Como ya se especificó en el glosario, S3 es un servicio de AWS de almacenamiento redundante y de alta disponibilidad.

- ✖ **Problemas:** Al igual que ocurre con GlusterFS, S3 no está diseñado para almacenar en él elementos que deban ser leídos o guardados con una alta tasa de velocidad, como los archivos de caché o sesiones de usuario. Altas latencias en este tipo de archivos disminuirían considerablemente la experiencia del usuario.

4.2.5.2.1.4 Redis

Redis [117] es un motor de base de datos en memoria RAM. Provee de una alta disponibilidad y baja latencia al estar los elementos guardados en dicha memoria. Permite la replicación Master/Slave

- ✖ **Problemas:** La memoria RAM, por sus características, tiene un alto coste. Si hacemos uso de este sistema para almacenar el contenido de la carpeta “media” necesitaríamos una gran cantidad, lo que repercutiría en el coste de dicho servicio.

4.2.5.2.1.5 Elección final: ElasticCache (Redis) + S3FS

El acceso a los archivos de caché y las sesiones de usuario debe de ser lo más inmediato posible, ya que de ello depende directamente la velocidad de carga de la web.

En cambio, el acceso a los archivos de imagen (directorio “media”) no es tan crítico ya que la carga se realiza desde el propio navegador cliente y no bloquea directamente la carga de la web por parte del servidor.

El servicio ElasticCache [118] forma parte de AWS y se puede configurar de forma que sea un sistema redundante. El servicio de S3 es un sistema ya de por sí altamente redundante.

Debemos tener en cuenta que S3 no se diseñó para ser usando como un sistema de archivos y que, además de ser un sistema en el que se paga por almacenamiento, también se paga por peticiones GET y PUT [119], lo cual, si va a ser usado como sistema de archivos con un alto tráfico, no es viable. Por ello, una vez el sistema esté en producción se habilitará el uso de un CDN (Content Delivery Network) [120], que cacheará los contenidos ubicados en la carpeta media (y por tanto en S3) y será él el responsable de servirlos a los navegadores, reduciendo las peticiones a dicho sistema a las actualizaciones diarias del CDN y a las subidas puntuales de archivos de imagen de la web.

Por lo tanto, la combinación de los dos sistemas nos proporciona un sistema que incluso mejora la velocidad del sistema NFS, pero manteniendo la alta disponibilidad, uno de los objetivos prioritarios de este proyecto.

Ésta será la combinación de sistemas que implementaremos, corrigiendo el diseño original que implementaba un servidor NFS.

4.2.6 Modificación del sistema. Sustitución NFS por instancia ElasticCache Redis + S3FS

El servicio Elastic Caché permite gestionar instancias Redis, las cuales se aprovisionan de la misma manera que las instancias EC2. Al igual que el servicio RDS, ElasticCache dispone del sistema de alta disponibilidad Multi-AZ, el cual genera una réplica del servidor principal y, en caso de caída del principal, AWS realiza un failover automático.

Aunque Magento viene preparado para comunicarse directamente con el servidor Redis, es recomendable instalar la extensión de PHP específica para ello, aumentando así el rendimiento. Lo haremos ejecutando el siguiente comando en nuestro servidor:

```
apt-get install php5-redis
```

S3 es el servicio de almacenamiento de alta redundancia de AWS. S3FS (S3 File System) basado en FUSE (FileSystem in UserSpace) nos permite montar un bucket (repositorio de almacenamiento) de S3 como si se tratara de una carpeta local de nuestro sistema. De esta forma, podemos almacenar la carpeta “media” en S3, evitando el uso de Redis para ello.

Aunque las imágenes no son un elemento que tenga que ser servido de forma tan inmediata como lo es la caché o la sesión de usuario, sí que es cierto que el hecho de que S3 no esté pensado para disponer de forma inmediata y con baja latencia de contenidos (menos aún si entre medias tenemos el sistema S3FS, que no es más que un sistema de sincronización entre S3 y nuestro sistema local) pueden generar lentitud en la carga de nuestra web. Además, si activamos en Magento la opción que unifica los archivos css y js en un solo archivo css y js, se puede acentuar el impacto del uso de S3FS, ya que dichos archivos unificados son almacenados en la carpeta media

Por ello se usará en conjunto con CloudFront, solución de CDN, de forma que S3 no será más que un sistema de almacenamiento unificado pero, en el fondo, quien se encargará de servir el contenido de la carpeta media será el CDN. De esta forma solucionamos el inconveniente que se producía al usar S3 como sistema de archivos y mejoramos la carga de nuestra web en todo el mundo, ya que CloudFront permite la distribución del contenido por nodos de distintos continentes.

A continuación el diagrama final que se implementará:

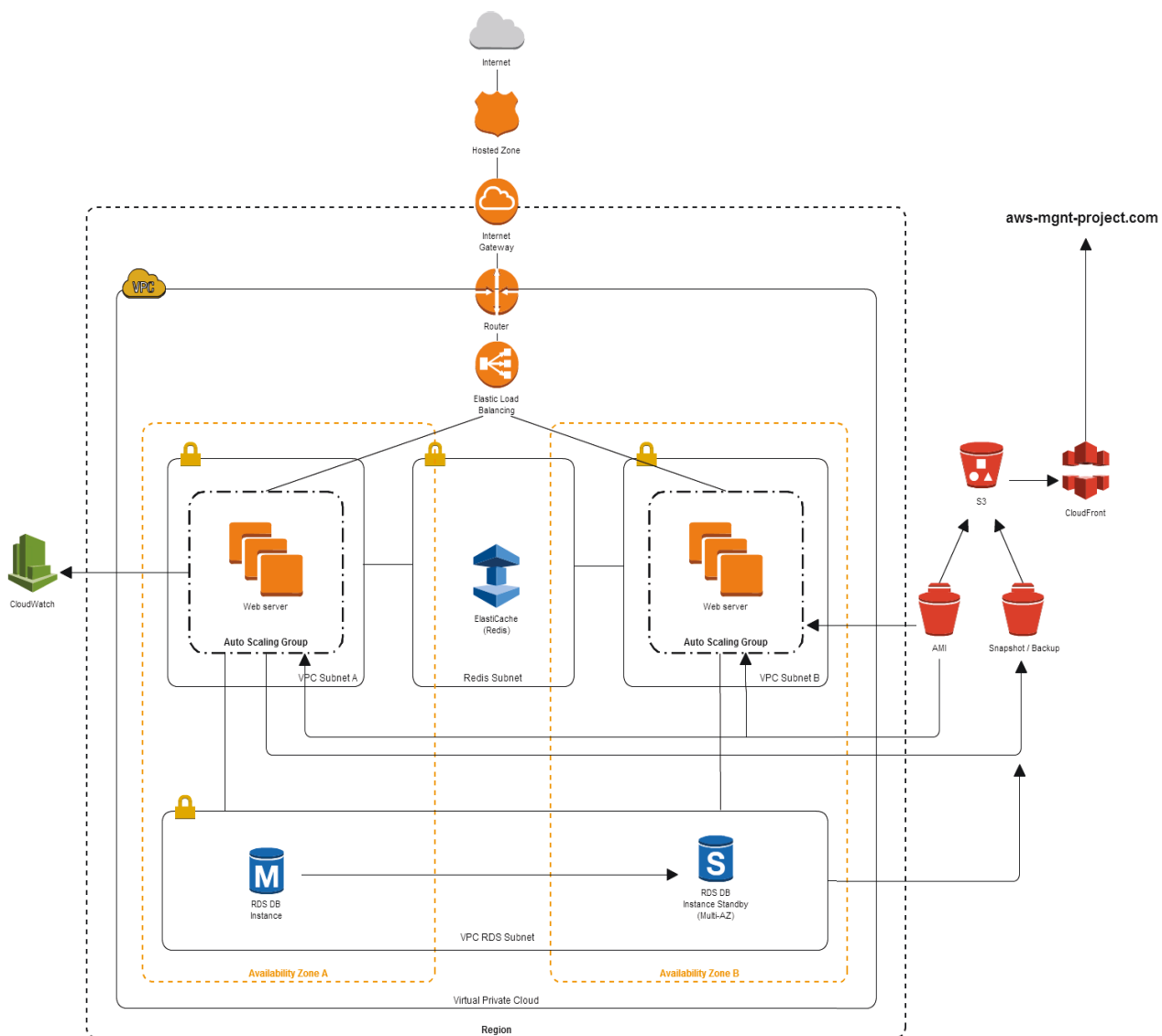


Ilustración 18: Diagrama modificado Sistema Autoescable y tolerante a fallos en AWS

4.2.7 ELB (Elastic Load Balancer)

El servicio ELB (Elastic Load Balancing) de AWS proporciona un balanceador el cual distribuye el tráfico entre distintos nodos con el fin de repartir la carga.

Podemos hacer uso de ellos como balanceadores externo o internos:

- ✓ **Externo** -> Gestiona el tráfico que tiene como origen internet o el exterior de nuestra VPC y como destino un nodo dentro de nuestra red. Todas las conexiones externas tienen que pasar por él para llegar a su destino.
- ✓ **Interno** -> Gestiona el tráfico que tiene como origen y destino un nodo interno de nuestra red. Puede usarse para control de carga en peticiones desde una o más instancias web a dos o más instancias de base de datos. El servicio ELB sólo permite balancear el tráfico que tiene como destino instancias EC2.

Es un servicio autoescalable en función de la demanda. En caso de aumentar el flujo de tráfico que debe gestionar, el sistema aumentará de forma automática y transparente los recursos, de forma que el servicio no se vea degradado.

El balanceo de la carga la realiza por medio de un algoritmo interno, de forma que sea lo más equitativa posible. Si activamos las “Cookie Stickiness” [121], el balanceador generará una cookie (con la duración que hayamos establecido) de forma que será capaz de identificar las peticiones realizadas por un mismo cliente y redirigirlas a la misma instancia. Ésta última funcionalidad está pensada para sistemas que no compartan la sesión de usuario, de forma que se intente mantener la misma para no dañar la experiencia del cliente. En nuestro caso no nos afecta al tener un directorio de sesiones compartido (Redis).

Debemos tener en cuenta que, al redirigir el ELB todas las peticiones, nos encontraremos con que tanto los logs de los servidores web como en el de Magento, no tendrán acceso a la IP del cliente, si no a la del balanceador, que es el que le está enviado la petición.

El servicio de ELB viene configurado para enviar la IP del cliente a través de la cabecera *X-Forwarded-For* [122] de forma que para conocer la IP real del cliente solamente tendríamos que acceder a dicha cabecera. Más adelante veremos cómo debemos configurar Magento y el servidor web para que obtengan la IP correcta.

AWS ha implementado distintas medidas de seguridad en su servicio ELB de forma que proteja el sistema balanceado de un incremento de tráfico inasumible o de ataques de denegación de servicio (DDoS) [123]. En el primer caso, si se detecta que el sistema balanceado no es capaz de asumir más carga, las nuevas peticiones son encoladas en el propio balanceador, evitando cargar así el sistema. Ante ataques DDoS, AWS implementa sus propias protecciones.

Además, dispone de dos servicios adicionales HealthyCheck y terminación SSL que nos ayudarán a tener un sistema más “sano” y ligero, respectivamente.

Más información [124]

4.2.7.1 Healthy Check

Función importante del balanceador de carga consistente en la comprobación del estado de las instancias. Periódicamente se realiza una petición GET a la ruta que hayamos configurado en este apartado y, en función de la respuesta, el test se dará o no por válido. En caso de no serlo, dejará de enrutar tráfico hacia la instancia problemática. En combinación con el servicio de AutoScalling, que veremos más adelante, permite ir más lejos y no sólo retira la instancia del grupo de reparto de carga si no que es eliminada (ELB) y sustituida (Autoscaling) por una nueva.

Es importante la correcta configuración de los parámetros del “Healthy Check” que se realiza, ya que de lo contrario, se puede dar lugar a un funcionamiento incorrecto.

El sistema nos permite configurar los siguientes parámetros:

- Protocolo. HTTP o HTTPS
- Puerto.
- Path. Permite elegir la ruta del test que se desea ejecutar.
- Tiempo de respuesta.
- Intervalo de tiempo entre tests.
- Número de tests fallidos antes de etiquetar la instancia como no saludable.
- Número de tests superados antes de etiquetar la instancia como saludable.

Normalmente es suficiente con un archivo cualquiera que nos devuelva una cabecera 200. No es válido apuntar a la raíz de un CMS ya que en numerosas ocasiones la primera respuesta enviada suele ser un 301, lo que se asumiría como un test fallido.

En nuestro caso hemos implementado un fichero llamado `healthycheck.php` que realiza una serie de comprobaciones, tales como asegurar el correcto montaje de la carpeta *media* en el sistema S3. Si el resultado de los test es satisfactorio, se genera una cabecera 200 OK. En caso contrario, se generará una cabecera de error, de forma que el balanceador asumirá que la instancia es problemática y no enrutará tráfico hacia ella. De esta forma no sólo comprobamos que el servidor web responde correctamente, si no que podemos comprobar que todo lo necesario para que la instancia preste un servicio adecuado funciona correctamente.

4.2.7.2 Instalación Certificado SSL

Otra característica importante del servicio ELB, es su capacidad de actuar como nodo final de la comunicación SSL (está siendo sustituida por el cifrado TLS por los graves fallos de seguridad [125] detectados).

La comunicación cifrada es necesaria para proteger los datos de los clientes en su envío desde el navegador web hasta los servidores de nuestro sistema. Generalmente ésta se ha de hacer hasta el propio servidor web pero, en nuestro caso, nos obligaría a tener instalado el certificado en todas las instancias web y soportando la carga extra de trabajo que provocan dichas conexiones.

Como hemos creado una red privada virtual, todo el tráfico que se genere dentro de ella es seguro, por lo que la conexión cifrada puede acabar en el balanceador, asumiendo éste la carga de trabajo extra que suponen el cifrado y descifrado de los datos.

Para ello, lo primero deberemos contratar un certificado con alguna de las múltiples autoridades de certificación (CA) [126] para acto seguido importarlo en el sistema de gestión de identidad y acceso de AWS (IAM), lo cual haremos a través de la consola AWS en cualquier instancia que despluguemos. Se puede ver un ejemplo a continuación:

```
#aws iam upload-server-certificate --server-certificate-name SSL-aws-mgmt-project --  
certificate-body file://my-certificate.pem --private-key file://my-private-key.pem --certificate-  
chain file://my-certificate-chain.pem
```

Respuesta:

```
{  
  "ServerCertificateMetadata": {  
    "Expiration": "2015-11-16T23:59:59Z",  
    "ServerCertificateId": "ASCAJ7ALXZDHWLUT6HUK2",  
    "ServerCertificateName": "SSL-aws-mgmt-project",  
    "Arn": "arn:aws:iam::457693362056:server-certificate/SSL-aws-mgmt-project",  
    "UploadDate": "2015-08-18T22:07:44.134Z",  
    "Path": "/"  
  }  
}
```

Una vez importando, durante la creación del balanceador debemos elegir los puertos de escucha, especificando que sean los puertos 80 y 443, ambos redirigidos al puerto 80 de las instancias web, ya que la conexión cifrada se acabaría en el balanceador.

4.2.7.3 ELB en instancias Web

Este servicio nos permite que, partiendo de una única dirección pública (dominio/IP pública) podamos tener más de una instancia servidor web procesando peticiones de forma paralela. Esta configuración posibilita tener al menos una instancia web en cada zona de disponibilidad, cumpliendo así con la alta disponibilidad pedida para este proyecto.

Siempre se pueden añadir más instancias, independientemente de su zona de disponibilidad, de forma que tres, cuatro o más servidores web procesen peticiones de forma paralela.

Para la puesta en marcha del ELB los parámetros a configurar serían:

- El entorno de red (Nuestro VPC).
- Puertos en los que nuestro ELB debe recibir las peticiones y redireccionarlas a las distintas instancias Web. Elegiremos los puertos estándar: 80 para el tráfico HTTP y 443 para el tráfico HTTPS. Ambos redirigidos al puerto 80 de las instancias web.
- Subredes o zonas de disponibilidad sobre las que debe balancear la carga.
- Grupo de seguridad. Se ha de seleccionar el grupo creado específicamente para el ELB Web.
- Configuración de los parámetros del Healthy Check.
- Instancias entre las que balancear la carga.

Es importante habilitar la opción “Cross-Zone Load Balancing” para permitir que el tráfico pueda ser balanceado entre instancias que se encuentren en distintas zonas de disponibilidad.

4.2.8 CloudWatch

El servicio CloudWatch permite supervisar varios de los recursos de AWS. Registra múltiples métricas de cada servicio, de forma que se puede monitorizar en tiempo real el estado de los mismos. El ejemplo uso más típico sería la monitorización de instancias EC2, de forma que podamos conocer la carga de la CPU, del disco o de la red de forma muy específica llegando, por ejemplo en el caso de los discos, a medir los Bytes escritos en disco o las operaciones de lectura.

En nuestro caso concreto, lo usaremos para el control de los recursos de las instancias EC2 y RDS.

La supervisión del servicio de RDS lo haremos con fines estadísticos, de forma que podamos prever una posible necesidad de escalado vertical u horizontal. En cambio, el control de las instancias EC2 es fundamental si queremos hacer uso del servicio de Autoescalado, ya que los datos obtenidos serán analizados para tomar decisiones sobre la necesidad o no de actuar en el sistema.

4.2.8.1 Alarmas

¿Cómo funciona? CloudWatch está permanentemente proporcionando métricas de las instancias asignadas a este sistema de monitorización. Se pueden definir alarmas que ejecuten acciones tales como el aviso por email en el momento en el que se superan o se rebajan determinados niveles en las distintas métricas. Por ejemplo, se puede crear una alarma que envíe un aviso en el momento que el consumo de CPU supere el 85% a lo durante de 5 minutos.

Éste es un servicio imprescindible en el entorno AutoScalling, ya que sirve de disparador de las diferentes configuraciones, de forma que se ejecuten políticas de incremento o decremento de instancias.

4.2.9 Autoscaling

El servicio de AutoScalling o Autoescalado de AWS mantiene nuestro sistema “vivo” de forma que, en función de la demanda, se adapta aumentando recursos para cubrir un pico de demanda o reduciéndolos en caso de no ser necesarios, conteniendo de esta manera costes.

El funcionamiento de este sistema se basa en los datos obtenidos a partir de dos servicios de AWS: CloudWatch y Elastic Load Balancing.

CloudWatch se encargará de monitorizar las instancias EC2 contenidas en el grupo de autoescalado. Basándose en los valores indicados en las alarmas configuradas evaluará si debe “disparar” alguna de ellas, provocando de esa manera que el servicio de autoescalado actúe aumentando o reduciendo recursos dependiendo de la alarma y de la configuración que hayamos hecho del mismo.

Si lo complementamos las alarmas de CloudWhatch con los test HealthyCheck del ELB, el servicio de autoescalado adquiere capacidad para tomar de decisiones sobre cada instancia de forma individual, pudiendo de esta manera actuar de forma específica sobre nodos problemáticos o inestables y no sólo sobre el grupo en su conjunto ajustando el número de nodos.

4.2.9.1 Autoescalado Web

En nuestro caso sólo usaremos el servicio de autoescalado para nuestras instancias web, pero puede ser usando para controlar entornos de todo tipo.

4.2.9.1.1 Creación de configuración de inicio

Para comenzar a hacer uso del servicio de AutoScalling de AWS, lo primero que debemos hacer es generar una configuración inicial que defina los parámetros de las instancias que van a ser usadas.

En primer lugar debemos elegir la imagen que se usará en las instancias, que será la creada en el apartado “4.2.4.2 Instancias Web”. Pasaremos a asignar los recursos, definidos por el tipo de instancia t2.medium, con las siguientes especificaciones:

- 2 x vCPU

- 4GB de Memoria RAM
- 8GB de almacenamiento en disco SSD. Si fuera necesario se puede incrementar el tamaño y/o realizar un aprovisionamiento personalizado de IOPS [127]

Elegiremos un nombre para identificar el perfil de configuración inicial, y habilitaremos el servicio de CloudWatch,

Esta es la configuración que será asignada a cada instancia que el servicio de autoescalado elija iniciar. Podemos crear diferentes configuraciones de inicio en previsión de futuras necesidades.

4.2.9.1.2 Creación del grupo de autoescalado

En este apartado crearemos grupos de escalado definiendo su estado inicial y configurando las políticas de aumento o disminución de recursos.

Podemos definir los grupos de forma que afecten a zonas de disponibilidad concretas o crear uno que englobe todas nuestra VPC. En nuestro caso, crearemos un solo grupo que afecte a toda nuestra VPC ya que en el momento de crear las instancias el sistema la asigna a cada zona de disponibilidad intentando que se queden lo más equilibradas posibles.

4.2.9.1.2.1 Configuración básica

Creemos un nuevo grupo de escalado eligiendo como configuración de inicio la creada anteriormente. Asignamos un identificador, el número de instancias inicial, la VPC que creamos y las dos zonas de disponibilidad disponibles en nuestra VPC.

Debemos especificar que el tráfico hacia nuestro grupo de autoescalado será suministrador por nuestro balanceador, el mismo que comprobará la salud de nuestras instancias.

Una instancia puede tardar varios minutos en estar operativa, por lo que debemos realizar los test oportunos para calcular de forma correcta un periodo de gracia antes de que los resultados de los HelthyCheck del balanceador sean tenidos en cuenta por el grupo de autoescalado. Si nos quedamos cortos, nos podríamos encontrar con que las instancias son eliminadas y sustituidas por unas nuevas sin haberles dado tiempo a estar operativas. Por defecto, se establecen 300 segundos.

Permitiremos que CloudWatch monitoree las instancias y pasaremos a definir las reglas de autoescalado de nuestro sistema.

Para evitar que un incremento desmesurado de demanda o un ataque pudieran provocar que nuestro sistema automático levante cientos de instancias, debemos especificar un límite máximo pero, teniendo cuidado en no ser excesivamente conservadores, ya que si no podríamos estar limitando las posibilidades de nuestro sistema de crecer ante una legítima demanda.

4.2.9.1.2.2 Configuración políticas de escalado

Las políticas de escalado definen las acciones a realizar y cuándo se deben hacer.

Por defecto se predefine una política que incrementa el número o el porcentaje de los nodos y una que realiza lo contrario, decrementa el número o porcentaje de nodos.

En cada política se puede establecer una alarma que provoca su ejecución si se cumple la condición definida. Por ejemplo, se puede establecer una alarma si el uso de CPU supera el 60% durante dos periodos de tiempo consecutivos de cinco minutos.

Se pueden crear más políticas dirigidas a incrementar o decrementar los nodos, sin necesidad de que exista una paridad. Esto puede ser útil cuando existen varias situaciones en la que se requiere una modificación de la cantidad de nodos.

Por ejemplo, puede existir la política de creación de nuevos nodos explicada anteriormente y crear una nueva, con el mismo objetivo de incrementar en uno el número de nodos, pero que sea disparada si el número de IOPS de lectura de disco supera un número determinado.

Al disparo de las alarmas se puede también asociar una notificación, de forma que podamos conocer el estado del sistema.

En nuestro caso crearemos las dos reglas por defecto. Una se encargará de crear un nuevo nodo en caso de que el uso de la CPU exceda el 60% durante un periodo de cinco minutos consecutivos. La regla contraria se encargará de reducir el número de instancias en una unidad en caso de que el uso de la CPU esté por debajo del 30% durante un periodo de 5 minutos consecutivos.

4.2.10 Grupos de seguridad

Los grupos de seguridad son una serie de perfiles que permiten asociar una determinada configuración de reglas de entradas/salida de firewall a los nodos que interactúen dentro de una misma VPC. Por tanto, dichos grupos se configuran por VPC y se asocian a cada nodo de forma individual.

Las conexiones salientes, por defecto, están completamente permitidas, al contrario de lo que ocurre con las conexiones entrantes. Si no especificamos los puertos que queremos que sean accesibles de nuestras instancias, no podrán recibir ningún tipo de conexión.

En nuestro caso necesitaremos cuatro grupos de seguridad:

- Grupo de seguridad para las instancias web
- Grupo de seguridad para las instancias RDS
- Grupo de seguridad para el balanceador web
- Grupo de seguridad para las instancias Redis

Ejemplo de política de seguridad entrante para el balanceador web:

| Type ⓘ | Protocol ⓘ | Port Range ⓘ | Source ⓘ |
|--------|------------|--------------|-----------|
| HTTP | TCP | 80 | 0.0.0.0/0 |
| HTTPS | TCP | 443 | 0.0.0.0/0 |

Ilustración 19: Política de seguridad entrante para el balanceador web

4.2.10.1. Grupo de Seguridad para las instancias RDS

Para las instancias de base de datos únicamente necesitamos permitir el acceso a través del puerto por defecto de MySQL, esto es, el puerto 3306. Sólo permitiremos conexiones desde nuestra VPC, por lo tanto asignaremos el CIDR de nuestra VPC: 10.0.0.0/24

4.2.10.2 Grupo de seguridad para el balanceador web

Para el balanceador web necesitaremos permitir las siguientes conexiones:

- Conexión HTTP -> Puerto 80. Redirige al puerto 80 en de las instancias web
- Conexión HTTPS -> Puerto 443. Redirige al puerto 80 de las instancias web.

En este caso debemos permitir conexiones desde cualquier lugar, ya que el balanceador es el encargado de recibir todas las peticiones de clientes y dirigirlas a las instancias web. Por lo tanto asignaremos el CIDR: 0.0.0.0/0

4.2.10.3 Grupo de seguridad para las instancias Web

Para las instancias web únicamente necesitamos permitir el acceso a través del puerto por defecto para HTTP, esto es, el puerto 80. Debemos recordar que todas las conexiones HTTPS finalizarán en el balanceador.

Sólo permitiremos conexiones desde nuestra VPC, por lo tanto asignaremos el CIDR de nuestra VPC: 10.0.0.0/24

4.2.10.4 Grupo de seguridad para las instancias Redis

Para que sea posible la comunicación con la instancia Redis hemos de crear un grupo de seguridad que permita el tráfico a través del puerto 6379.

Sólo permitiremos conexiones desde nuestra VPC, por lo tanto asignaremos el CIDR de nuestra VPC: 10.0.0.0/24

4.2.11 Route53

Route53 es el servicio web DNS escalable y de alta disponibilidad de AWS. Al estar integrado dentro de su sistema, nos permite total compatibilidad con el mismo e incluso el uso de dominios internos que no pueden ser resueltos desde el exterior.

Dicho sistema es usado de forma interna por AWS para la creación de las direcciones o hostnames de sus servicios y para, en caso de actuación del sistema Multi-AZ por caída de una zona de disponibilidad, redirigir las peticiones de cada sistema a su réplica en otra subred.

Al igual que hemos visto en el servicio ELB, Route53 implementa la herramienta HealthyChecks, que permite verificar si los dominios que se le especifiquen están “vivos”, es decir, se recibe algún tipo de respuesta, pudiendo implementar notificaciones en caso de problemas.

En nuestro caso lo usaremos Route53 de tres maneras diferentes:

- Por un lado gestionará nuestro dominio (<http://aws-mgnt-project.com>) que apuntaremos mediante CNAME [128] al balanceador web.
- Crearemos un dominio de uso interno para la gestión de todos los servicios que disponemos, de forma que, partiendo de dicho dominio, crearemos un subdominio para la instancia de base de datos y otro para la instancia Redis. De esta forma, si en algún momento se necesita realizar algún proceso de mantenimiento que implique la parada de uno de estos servicios y, por tanto, la posibilidad de que su “endpoint” cambie, no tendremos que más que asociar la nueva dirección en Route53, no teniendo que realizar cambio alguno en la configuración de nuestro Magento.
- Por último, Route53 será usado de forma automática por AWS para gestionar los procesos de failover en caso de que se detecten problemas en las instancias protegidas por el sistema MultiAZ, es decir, las instancias de base de datos (RDS) y Redis (ElasticCaché)

4.2.12 CloudFront

CloudFront es un servicio Web de entrega de contenido (CDN) con baja latencia y altas velocidades de transferencia que permite la replicación del contenido por servidores de todo el mundo, de forma que los visitantes de nuestra web se descarguen el contenido del servidor más cercano a su ubicación.

Se usa principalmente para el almacenamiento de ficheros CSS, JS e imágenes, aunque también permite lidiar con contenido dinámico. En nuestro caso lo usaremos para servir el contenido de /media, /css y /js de nuestro sistema, es decir, imágenes, hojas de estilo y archivos JavaScript.

Al funcionar como una caché debe permitirse la invalidación de los ficheros para que, en caso de que se necesitasen realizar cambios, éstos sean propagados lo antes posible. Cada elemento es invalidado una vez expira su tiempo de vida indicado por el servidor web. Si no se ha indicado nada, automáticamente su tiempo de vida es de 24h. En caso de que necesitemos invalidar uno o varios objetos, podemos solicitar una invalidación de los mismos a través del panel de CloudFront o haciendo uso de la API.

Hay ciertas configuraciones que debemos tener en cuenta, como el uso de HTTPS cuando sea necesario, habilitar el uso de la cabecera OPTIONS [129] para las peticiones CORS [130], o el reenvío de las cabeceras de caché especificadas por el servidor web, de forma que no sólo CloudFront pueda conocer su tiempo de vida, sino también el navegador usado por el visitante, de forma que sólo solicite los archivos necesarios.

4.3 Desarrollo Magento

Como ya hemos hablado anteriormente, Magento es un sistema gestor de contenidos orientado al comercio electrónico. Hasta ahora hemos diseñado e implantado toda la arquitectura sobre la que instalaremos Magento y que le dotará de alta disponibilidad y capacidad de crecimiento y decrecimiento en función de la demanda.

El siguiente paso es la configuración de dicho software de forma que nos permita englobar tres tiendas distintas, cada una orientada a un cliente distinto (Europeo, Asiático o Americano) con sus propias configuraciones, productos, diseño, precios, etc todo ello administrado desde un solo panel. Para ello necesitaremos realizar una serie de configuraciones en el servidor web, ya que será éste el encargado de indicar a Magento la procedencia geográfica del cliente, de forma que le muestre la tienda adecuada.

Una de las motivaciones para la configuración de tres tiendas distintas en Magento es la política logística, consistente en la descentralización de los almacenes en zonas estratégicas de forma que seamos capaces de servir los productos vendidos en el menor tiempo posible y a un precio razonable en la mayor parte del mundo. Ello no quiere decir que se vaya a enviar a todos los países del mundo, ni que el hecho de que la tienda "Europa" se llame así por la ubicación del centro logístico signifique que sólo se envía a Europa. La relación de países a los que se enviará será una decisión interna del cliente y los envíos se harán desde el centro logístico no ya más cercano, si no el que conlleve menor coste de envío.

Al poder controlar todos los pedidos realizados en las tres tiendas y su contabilidad desde un solo sistema/panel, Magento nos servirá como solución de contabilidad y gestión de la empresa, pudiendo realizar facturas y devoluciones. De esta forma, simplificamos la gestión, delimitamos el aprendizaje a un solo sistema y ahorramos costes.

4.3.1 Instalación

La instalación inicial de Magento y su configuración, normalmente se suele realizar en un entorno local (haciendo uso de XAMPP [131] o similar) o en un entorno de desarrollo en un servidor en la nube (haciendo uso de AWS o similar). En cualquier caso, es preferible intentar reproducir el entorno que tendremos en preproducción y producción, con el fin de encontrarnos el menor número de problemas.

La herramienta con la que podremos llevarlo a la práctica de la mejor forma tanto si decidimos realizar la instalación en local o en la nube, es Vagrant.

Vagrant es una potente herramienta para la creación y configuración de entornos de desarrollo virtuales. A través de su fichero de configuración podremos definir los parámetros de nuestro entorno, el sistema operativo (Ubuntu, Debian, CentOS, etc), los paquetes de software necesarios (Apache, PHP y sus dependencias, MySQL, etc).

Podremos elegir si queremos que la máquina sea local (VirtualBox) o en la nube, en cuyo caso, proporcionando las credenciales de nuestro proveedor en la nube (AWS, DigitalOcean [132], etc), la máquina será creada remotamente.

Una vez tengamos todo configurado, con el comando *vagrant up* la herramienta se encargará del despliegue de la máquina y de realizar toda la configuración contenida en el fichero que hemos creado.

Puphpet es una herramienta online que nos ayuda a crear el fichero de configuración de Vagrant de una manera muy sencilla. Podremos elegir el sistema operativo, el tipo de servidor web, crear host virtuales, elegir servidor de bases de datos, crear las bases de datos y los usuarios, los paquetes de software a instalar y un largo etc. Todo lo necesario para generar un entorno lo más parecido a preproducción o producción. Una vez seleccionada la configuración, se descargará el fichero de configuración que Vagrant usará para crear la máquina.

4.3.1.1 Control de versiones

En caso de que vayamos a efectuar un desarrollo a medida o modificaciones en Magento (sea desarrollando algún módulo o modificando el diseño o funcionalidad de la plantilla), no debemos olvidar hacer uso de un sistema de control de versiones como GIT.

Esto permite que múltiples personas puedan desarrollar al mismo tiempo, tener un control de versiones con el que retroceder a una versión estable de nuestro código y, finalmente, hacer *pull* del repositorio (copiar nuestro código) en el servidor en preproducción o producción.

4.3.2 Configuración e implantación

En este apartado se tratarán los procedimientos necesarios para la configuración del sistema Magento y su implantación en el sistema hardware creado anteriormente.

4.3.2.1 Configuración con balanceador web

4.3.2.1.1 Dirección IP origen

Uno de los problemas con los que nos encontramos al hacer uso de un balanceador, es que para los servidores web que reciben las peticiones (es decir para Magento en este caso) la ip que origina la petición es la del balanceador. Esto supone dos problemas.

- Por un lado no podemos identificar la IP de origen para estadísticas propias de Magento o incluso comprobaciones de fraude. En los logs del servidor web tampoco aparecería la IP real de las conexiones.
- El segundo problema, y más importante en nuestro caso, es que al no conocer la IP de origen Magento no podría decidir qué tienda debe mostrar al cliente por su ubicación, por lo que mostraría la tienda por defecto que, en caso de no ser la adecuada, impediría realizar la compra al cliente al no realizar envíos a su país.

Los ELB de Amazon por defecto envían de la cabecera “X-Forwarded-For” en la que se inserta la IP de origen de la petición. La “X” inicial nos indica que no es una cabecera original del protocolo HTTP (RFC1945 [133], RFC2616 [134], RFC2774 [135]) si no una cabecera personalizada, que, como se indica en el RFC2047 [136], deben ser identificadas por comenzar con la letra “X-”

De esta manera, en nuestro servidor web podremos recoger dicha cabecera y conocer la IP real de las peticiones. Para que Magento sepa que debe leer esta cabecera para conocer la IP real, debemos añadir las siguientes líneas al archivo de configuración de Magento (MAGENTO_DIR/app/etc/local.xml):

```
<remote_addr_headers>
  <header1>HTTP_X_FORWARDED_FOR</header1>
</remote_addr_headers>
```

Además, para que en los logs de nuestro servidor apache quede correctamente registrada la ip de origen, debemos modificar el formato del log en la configuración de nuestro “virtual-host”. Otra opción más sencilla es hacer uso del módulo “mod_remote_ip” [137].

4.3.2.1.2 Conexión segura (SSL)

Como ya se comentó al hablar sobre el servicio de ELB, en el balanceador finalizará la conexión SSL. De esta forma nos evitamos tener que realizar la configuración SSL en cada uno de los nodos web y, además, les liberamos de la carga extra que supone una comunicación cifrada.

Ello nos plantea un problema adicional y que es que en Magento debemos configurar la conexión segura para que, en las páginas necesarias (checkout, login,etc), fuerce el uso de este tipo de conexión por parte del navegador del cliente. El problema se generaría al enviar el navegador la nueva solicitud de conexión segura a Magento, ya que, al terminar este tipo de conexión (puerto 443) en el balanceador, la conexión continuará a los servidores web a través de conexión no segura (puerto 80). Esto provocaría problemas con Magento ya que no aceptaría la petición y forzaría volver a usar SSL, lo cual generaría un bucle infinito.

Para solventar este problema, debemos “engañar” a Magento cuando detectemos que se inicia una conexión segura. Para ello, situaremos el siguiente código en el fichero index.php antes de la llamada a Magento:

```
if( isset($_SERVER['HTTP_X_FORWARDED_PROTO']) ) {  
    if($_SERVER['HTTP_X_FORWARDED_PROTO'] == 'https') {  
        $_SERVER['HTTPS'] = 'on';  
        $_SERVER['SERVER_PORT'] = 443;  
    }  
}
```

La cabecera “HTTP_X_FORWARDED_PROTO” indica el protocolo de conexión usado en la conexión del cliente al balanceador. Por lo tanto, si dicha cabecera existe y la conexión fue realizada haciendo uso del protocolo HTTPS, cambiaremos el contenido de las cabeceras “HTTPS” y “SERVER_PORT”, a “ON” y “443” respectivamente. De esta manera, aunque no sea así, Magento creará que la conexión es segura y responderá a la misma.

4.3.2.2 Geolocalización del cliente

Una de las opciones para que Magento muestre la tienda asociada a la localización del usuario es instalar un módulo de GeoLocalización en Magento, el cual, una vez que se recibe la petición de conexión, analiza la IP y ordena a Magento mostrar la Website adecuada.

En este caso lo vamos a hacer instalando el módulo de localización directamente en el servidor web. De esta forma será él quien evalúe la Website que se deba cargar y directamente genere la llamada adecuada a Magento, ahorrando tiempo al pedir a Magento directamente lo que necesitamos.

Instalamos el módulo:

```
sudo apt-get install libapache2-mod-geoip
```

Abrimos el fichero de configuración:

```
/etc/apache2/mods-enabled/geoip.conf
```

Y escribimos los comandos necesarios para indicar la localización de la base de datos de países (que nos podremos descargar de forma gratuita maxmind.com) y activar el módulo:

```
GeoIPDBFile /path/to/database.dat
```

```
GeoIPScanProxyHeaders On
```

El siguiente paso es crear las distintas “websites” en la configuración de Magento, su tienda y las “vistas de tienda” (una por cada idioma que queramos mostrar). A cada “website” le habremos asignado un código identificador que usaremos para generar la llamada a Magento.

Para ello modificaremos el archivo “index.php” sustituyendo:

```
$mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : "";
$mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'store';
Mage::run($mageRunCode, $mageRunType);
```

Por:

```
/* Switch countries */
```

```
// Website Europa
$europa = array(
    'ES' => True,
    'DE' => True,
    'PT' => True,
    'FR' => True,
    'PL' => True,
    'IT' => True,
    'MO' => True,
    'ZA' => True,
);
```

```
// Website America
$america = array(
    'US' => True,
    'MX' => True,
    'BR' => True,
    'CL' => True,
    'AR' => True,
    'CO' => True,
    'PE' => True,
    'PA' => True,
);
```

```
// Website Asia
$asia = array(
    'CN' => True,
    'JP' => True,
    'HK' => True,
    'TH' => True,
    'VN' => True,
    'RU' => True,
    'IN' => True,
    'ID' => True,
);
```

```
$code = $_SERVER['GEOIP_COUNTRY_CODE'];
```

```
switch (true) {
    case isset($europa[$code]):
```



```

    $mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : 'europa';
    $mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'website';
    break;
case isset($america[$code]):
    $mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : 'america';
    $mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'website';

    break;
case isset($asia[$code]):
    $mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : 'asia';
    $mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'website';
    break;
default:
    $mageRunCode = isset($_SERVER['MAGE_RUN_CODE']) ? $_SERVER['MAGE_RUN_CODE'] : 'europa';
    $mageRunType = isset($_SERVER['MAGE_RUN_TYPE']) ? $_SERVER['MAGE_RUN_TYPE'] : 'website';
}

Mage::run($mageRunCode, $mageRunType);

```

De esta forma Magento directamente comenzará a carga la website adecuada para el cliente. Los distintos países se irán añadiendo según el cliente vaya teniendo capacidad de envío a los mismos.

Además, para cada tienda crearemos un árbol de categorías diferente, lo que permitirá tener una categorización diferente y adecuada a los productos que se vendan o al diseño de la web que, también, será diferente para cada tienda. De esta forma se busca adaptar cada tienda a los gustos de cada mercado.

4.3.2.3 Configuración interna de Magento

Se deberán configurar todos los elementos necesarios para el correcto funcionamiento de la tienda, como puede ser los datos de la empresa, productos, categorías, las distintas páginas estáticas que queramos añadir (contacto, páginas legales, etc), métodos de pago y envío y múltiples configuraciones más. Pasaremos a detallar aquellas que por la naturaleza del proyecto hay que destacar o que necesitan una atención especial.

4.3.2.3.1 Monedas

En cada una de las tiendas, se habilitarán diferentes monedas, de forma que, por ejemplo, en la tienda de Europa, los consumidores puedan ver los precios en varias monedas (Euros, Libras, Coronas suecas, Corona danesa, etc). Siempre debe haber una moneda base que es sobre la que se calcula la conversión a la moneda elegida y sobre la que finalmente se realizará el pago.

Para que el cambio sea lo más real posible, se conectará Magento con un servicio externo que mantenga los tipos de conversión actualizados. Magento tiene instalado su propio sistema, pero podemos ajustarlo de forma que se conecte con el servicio que nosotros queramos o consideremos más fiable. En nuestro caso lo conectaremos con el servicio de Google Finance. De esta forma, todas las noches, mediante el servicio de cron se descargará y procesará un archivo CSV [138] desde Google.

4.3.2.3.2 Idioma

Para cada tienda, podremos elegir tener diferentes idiomas, en función de las necesidades concretas de cada una de ellas. Por ejemplo, para la tienda de América podremos tener disponibles el Inglés, Español, Francés y Portugués. En la configuración de la tienda, debemos crear una “Vista de tienda” o “Store View” para cada idioma y asignar a cada una de ellas el idioma elegido.

Por último deberemos descargarnos la traducción oficial bien haciendo uso de algún paquete antiguo todavía disponible para descarga o bien a través de un módulo.

Es recomendable, sobre todo si no usamos la plantilla original de Magento, contratar a personal que nos revise las traducciones, corrigiendo lo que no sea correcto y añadiendo las traducciones que puedan ser necesarias.

4.3.2.3.3 Impuestos

Magento nos permite configurar los impuestos de diferentes países y varios impuestos por país. Se debe configurar acorde con el país en el que está registrada la empresa que genera la venta.

En casos como el que nos atañe, en el que se venden productos en múltiples países (y muchos de ellos fuera de la Unión Europea), se toma la decisión de crear delegaciones de la empresa en diferentes países, lo cual puede ayudar a solventar algunos problemas con los servicios de aduanas. En lo que nos afecta a Magento, éste nos permite elegir que, por ejemplo, la website “Asia” tenga una configuración de impuestos, y de empresa completamente diferente y acorde a los datos de una delegación que se pudiera establecer en algún país asiático.

4.3.2.3.4 Métodos de pago

Si fuese necesario, en cada website se podrá elegir tener activos distintos métodos de pago o que éstos tengan diferentes configuraciones, de forma que nos puede no interesar el pago por Western Union en Europa, pero sí en América. O, si para la website “Asia” hemos creado una delegación de nuestra empresa en China, podremos configurar los métodos de pago ya existentes en otras websites pero con los datos concretos de la delegación en China.

4.3.2.3.5 Sistema de envío

Cada tienda lógicamente debe limitar sus países de envío, por un lado para no compartírselos con el resto de tiendas y, por otro lado, porque si un país no pertenece a una tienda es debido a que no se quiere permitir la venta a ese país por diferentes motivos.

Dado que nuestro sistema tendrá la capacidad de vender alrededor de todo el mundo, necesitamos un sistema logístico mundial para evitar los altos costes de envío que pagaría, por ejemplo, un cliente tailandés si situáramos el almacén en Europa y queremos un tiempo de envío razonable (menos de una semana).

Para ello utilizaremos una empresa de gestión logística a nivel mundial, ubicando nuestros productos en sus almacenes en cada zona geográfica donde queramos realizar ventas (Europa, Asia y América). Mediante la conexión de nuestra tienda Magento a su central, los pedidos ya pagados serán automáticamente procesados y enviados, informando a Magento (y por tanto al cliente) del número de seguimiento de dichos envíos en el momento en el que estén disponibles.

De esta forma, tendríamos un sistema de venta mundial totalmente automatizado y que no necesitaría de intervención humana para que un pedido realizado por un cliente fuese procesado, enviado por el almacén más cercano y entregado al destinatario.

4.3.2.3.6 Control de Stock

Al elegir un sistema de envío descentralizado (con más de un almacén) nos encontramos antes el problema de la gestión del stock. Magento por defecto no permite más de un almacén, por lo que existirían dos opciones:

Delegamos el control del stock en el operador logístico. De esta forma desde Magento no tenemos control algún del stock real, pero sí desde el sistema del operador.

Instalamos un módulo en Magento que nos permita añadir la funcionalidad extra de permitir varios almacenes, uno por website. Esto nos permitiría que Magento estuviese informado del stock existente de cada producto en cada almacén. De esta forma no tenemos que estar controlando varios sistemas y, además, en caso de no existir stock de algún producto, Magento lo marcaría como tal. El operador logístico se encargaría de actualizar periódicamente Magento con el stock existente de cada producto en cada almacén.

4.3.2.3.7 Cron

El cron [139] es el sistema de programación de tareas de Unix. Permite la ejecución de tareas programadas en un determinado momento con o sin repetición.

Para el correcto funcionamiento de Magento, se necesita la ejecución del script cron.php ubicado en la raíz de Magento. Este script se encarga de realizar las tareas programadas de Magento, entre las que pueden estar el envío de correos, limpieza de base de datos, sincronización de la tabla de conversiones de moneda y un largo etcétera. Debe ser ejecutado cada pocos minutos.

4.3.2.3.8 Configuración de envío de emails

Deberemos realizar la configuración de los emails para cada websites (direcciones corporativas, diseño, mensaje de texto, política de envío, etc). A través de la función de “Correos transaccionales”, podremos especificar el diseño, contenido e idioma de los emails para cada Tienda.

Para realizar el envío de emails, debemos tener un servidor SMTP [140] que se ocupe de ello. Podemos usar un proveedor de correo externo o hacer uso del servicio SES (Simple Email Service), el cual nos proporciona un servidor SMTP para realizar envíos masivos de correo de forma controlada.

Esta segunda opción nos permite tener la seguridad de tener un servicio de envío de emails seguro, escalable y controlado. En cualquier caso, deberemos de contar con un servicio de correo electrónico independiente (contratado con un proveedor externo o con el servicio de AWS WorkMail [141]) ya que SES únicamente realiza envíos de correo.

Una vez decidido el sistema a usar, deberemos configurar nuestro servidor para que se conecte con el sistema elegido:

- Instalar en nuestro servidor de un MTA (Mail Transport Agent) [142] como Exim [143] y configurarlo para que se conecte al servidor SMTP elegido.
- Instalar un módulo en Magento que haga las funciones de un MTA, conectándose directamente al servidor SMTP que configuremos.

Consideramos la segunda opción como la más recomendable.

4.3.2.3.10 Sesiones de usuario

Las cookies de sesiones de usuario se crean en el momento en el que un cliente accede a la tienda y se guardan en el propio servidor y en el navegador del cliente. Esto permite a Magento realizar un seguimiento del cliente, las páginas que visita, si está logueado o no, si ha añadido artículos al carro de la compra, etc.

Como comentamos en apartados anteriores, es importante que estas cookies estén accesibles para todos los nodos web, ya que si no, se podría dar el caso de que el balanceador redirigiera al cliente a un nodo distinto que no contiene su sesión, por lo que si estaba comprando, perdería el contenido del carro de la compra.

No es válido el uso de “sticky sessions” en el balanceador ya que al estar en un entorno autoescalable, en cualquier momento cualquier máquina puede ser eliminada.

Para este sistema se ha elegido una base de datos Redis, la cual hace uso de un almacenamiento basado en tablas de hashes [144] que almacena en memoria RAM. Esto nos permite alta disponibilidad y gran velocidad.

Desde la versión 1.8 Magento viene con soporte oficial para la conexión a Redis. Un ejemplo de código para especificar a Magento que debe guardar las sesiones en el servidor redis sería el siguiente:

```
<session_save>db</session_save>
<redis_session>
  <host>redis.internal-aws-mgmt-project.com</host>
  <port>6379</port>
  <password></password>
  <timeout>2.5</timeout>
  <persistent></persistent>
  <db>1</db>
  <compression_threshold>2048</compression_threshold>
  <compression_lib>gzip</compression_lib>
  <log_level>1</log_level>
  <max_concurrency>6</max_concurrency>
  <break_after_frontend>5</break_after_frontend>
  <break_after_adminhtml>30</break_after_adminhtml>
  <first_lifetime>600</first_lifetime>
  <bot_first_lifetime>60</bot_first_lifetime>
  <bot_lifetime>7200</bot_lifetime>
  <disable_locking>0</disable_locking>
  <min_lifetime>60</min_lifetime>
  <max_lifetime>2592000</max_lifetime>
</redis_session>
```

A partir de este momento, no deberían volver a guardarse las sesiones en *magento/var/sessions*

4.3.2.3.11 Optimización de Magento

De cara a la mejora del rendimiento de Magento, debemos habilitar ciertas características:

- Merge JS y CSS -> Agrupa en un solo fichero CSS todos los archivos CSS necesarios para la plantilla y el backoffice de Magento. Lo mismo con los archivos JS. De esta forma se reducen de forma considerable las peticiones de conexión para la descarga de los ficheros.
- Habilitar “Flat Catalog Category” y “Flat Catalog Product” -> Activa una caché de productos y categorías en Magento. Debido al uso de una estructura EAV (Entity Attribute Value) [145] en la base de datos de Magento, la lectura de todos los elementos que componen un solo producto implica el acceso a múltiples tablas de la base de datos. Este diseño responde a la necesidad de organizar de forma congruente miles de productos, pero implica el uso de consultas SQL muy grandes, con la consiguiente ralentización del sistema.
Al habilitar el uso de “flat resource” Magento crea una tabla en la que mantiene los productos y sus atributos. De esta manera las consultas necesarias para la lectura de productos son muy simples y aligeran el sistema.
- Habilitar optimizador de código PHP -> De cara a mejorar la velocidad de Magento, es recomendable la instalación de algún optimizador o caché de código PHP (OPC [146] o APC [147], por ejemplo). En caso de no ser posible, Magento dispone de un “Compilador” el cual puede ayudar a mejorar el rendimiento del mismo.
- Habilitar la caché de Magento y hacer uso de un CDN.

4.3.2.3.11.1 Caché

Un sistema de caché almacena de forma temporal datos que ya han sido procesados, es decir, una vez que se ha procesado una página, guarda una copia, de forma que la siguiente vez que se pida ese mismo elemento no habrá que procesarlo si no enviarlo directamente. De esta forma tenemos un sistema mucho más rápido al no tener que procesar todas las peticiones y permite afrontar mayor carga de trabajo.

Magento está preparado para una implementación de dos niveles de caché, siendo la primera una caché rápida y la segunda una caché más lenta (disco o base de datos).

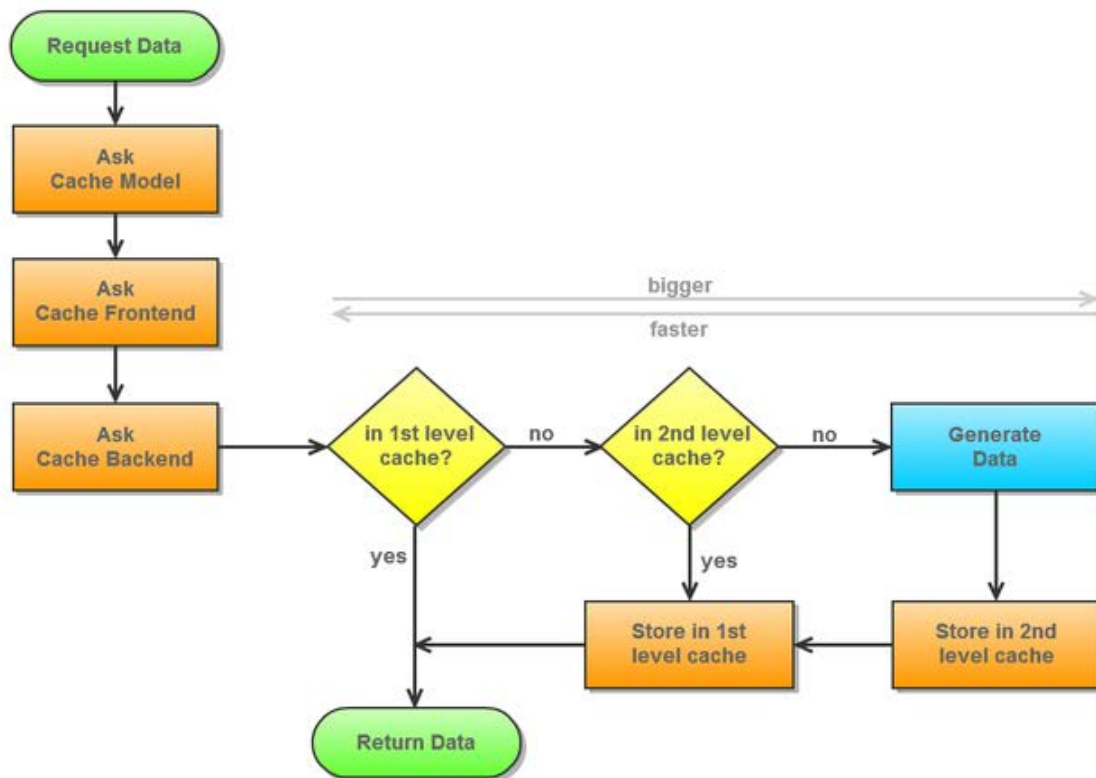


Ilustración 20: Funcionamiento caché dos niveles Magento

Desde la versión 1.8 Magento tiene incorporado el código necesario para hacer uso de La base de datos Redis como caché. Al tener ésta soporte también para tags (normalmente ubicados en la caché lenta), podemos fusionar los dos niveles de caché en uno sólo, eliminando el nivel de caché “lenta”.

En grandes despliegues de Magento, se puede hacer uso de Redis como nivel de “caché lenta” y de Memcached [148] como “caché rápida” (no soporta tags).

En nuestro caso, para habilitar Redis como único nivel de caché, debemos añadir el siguiente código:

```

<cache>
  <backend>Cm_Cache_Backend_Redis</backend>
  <backend_options>
    <server>redis.internal-aws-mgnt-project.com</server>
    <port>6379</port>
    <persistent></persistent>
    <database>0</database>
    <password></password>
    <force_standalone>0</force_standalone>
    <connect_retries>1</connect_retries>
    <read_timeout>10</read_timeout>
    <automatic_cleaning_factor>0</automatic_cleaning_factor>
    <compress_data>1</compress_data>
    <compress_tags>1</compress_tags>
    <compress_threshold>20480</compress_threshold>
  </backend_options>
</cache>

```

```

    <compression_lib>gzip</compression_lib>
    <use_lua>0</use_lua>
  </backend_options>
</cache>

```

Acto seguido debemos habilitar y refrescar la caché de Magento.

4.3.2.3.11.2 Flat Catalog Product / Flat Catalog Category

Para el almacenamiento de clientes, pedidos, productos y categorías Magento hace uso del modelo de datos EAV o entidad-atributo-valor, el cual, permite la adición de todos los atributos que sean necesarios en cada caso ni necesidad de la modificación de las tablas.

Por ejemplo, por defecto podemos tener una serie de atributos que definen un producto (Entity) pero tenemos la necesidad de añadirle uno más, como puede ser la talla. En un modelo entidad-relación, al estar predefinido en la creación de la tabla las columnas y, por tanto, los atributos de producto, deberíamos ampliar la tabla (ALTER TABLE) lo cual, en el momento en el que empezamos a tener una gran cantidad de registros, es inestable y costoso en tiempo. En cambio, si hacemos uso del modelo EAV, simplemente tendríamos que crear dicho atributo en la tabla de atributos (Attribute) relacionándolo con la entidad en cuestión y, en la tabla de valores (Value) crear un nuevo registro relacionado con el atributo nuevo e insertando su valor.

De esta forma crear atributos nuevos es tan sencillo como añadir filas a varias tablas. Desventaja, este modelo de datos tiene un problema de rendimiento consecuencia del uso de varias tablas: Las consultas SQL para leer un producto/categoría/cliente pasan de ser sencillas a enormes o incluso monstruosas, lo que tiene como consecuencia una pérdida de rendimiento y lentitud del sistema, algo de lo que se lleva tiempo acusando a Magento. Además Magento separa los “valores” según su tipo (varchar, text, int, etc) lo cual incrementa considerablemente el número de tablas a consultar.

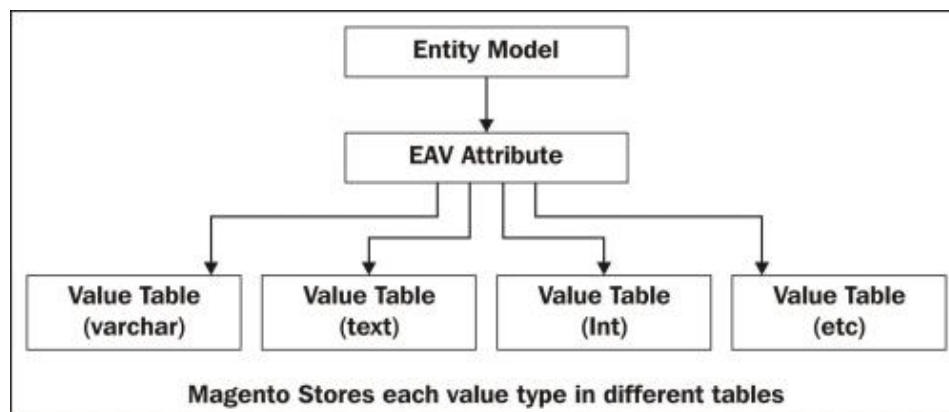


Ilustración 21: Model EAV en Magento [149]

Para solventar esto, se ha implantado la funcionalidad “Flat resource” mediante la cual Magento crea una nueva tabla por entidad (sólo para productos y categorías) agrupando en ella todos los datos necesarios por cada elemento. De esta forma la consulta de, por ejemplo, un producto, se vuelve a poder resolver leyendo una sola tabla. Esto no acaba con el modelo EAV, simplemente, partiendo de los datos almacenados en dicho modelo, se crean y mantienen una serie de tablas con el modelo de datos tradicional de forma que facilite su consulta.

Del mantenimiento de estas tablas se encarga el cron de Magento, que debe ser ejecutado periódicamente de forma que estas tablas se mantengan actualizadas.

Para activar esta funcionalidad debemos activar las opciones “Use Flat Catalog Product” y “Use Flat Catalog Category” y, acto seguido, reindexar dichas tablas de forma que se queden preparadas para su uso. A partir de entonces será el cron el que se encargue de mantenerlas actualizadas.

4.3.2.3.11.3 CDN

Como ya explicamos en apartados anteriores, haremos uso del servicio de CloudFront de AWS para generar una red de contenido distribuido a lo largo del mundo que permita la rápida descarga de los elementos estáticos (css, js e imágenes).

Una vez configurado y activado el servicio debemos configurar Magento para que genere las URL de estos contenidos apuntando a nuestra red de distribución y no a él mismo. Esto lo haremos en Sistema/Configuración/Web modificando los siguientes campos tanto en conexión insegura como segura:

- URL Base de Skin
- URL Base de Media

Podemos ver un ejemplo de cómo quedaría la configuración tanto para la conexión segura como para la insegura:

| Inseguro | | |
|--|--|--------------|
| URL Base | <input type="text" value="http://aws-mgnt-project.com/"/> | [STORE VIEW] |
| URL Base de Link | <input type="text" value="{{unsecure_base_url}}"/> | [STORE VIEW] |
| URL Base de Skin | <input type="text" value="http://d23je69vyjsf8o.cloudfront.net/skin/"/> | [STORE VIEW] |
| URL Base de Media | <input type="text" value="http://d23je69vyjsf8o.cloudfront.net/media/"/> | [STORE VIEW] |
| URL Base de JavaScript | <input type="text" value="{{unsecure_base_url}}js/"/> | [STORE VIEW] |
| <p>▲ Warning! When using CDN, in some cases JavaScript may not run properly if CDN is not in your subdomain</p> | | |

Ilustración 22: Configuración CDN Magento para conexiones HTTP

| Seguro | | |
|------------------------|---|--------------|
| URL Base | <input type="text" value="https://aws-mgnt-project.com/"/> | [STORE VIEW] |
| | ▲ Make sure that base URL ends with '/' (slash), e.g. http://yourdomain/magento/ | |
| URL Base de Link | <input type="text" value="{{secure_base_url}}"/> | [STORE VIEW] |
| | ▲ Make sure that base URL ends with '/' (slash), e.g. http://yourdomain/magento/ | |
| URL Base de Skin | <input type="text" value="https://d23je69vyjsf8o.cloudfront.net/skin/"/> | [STORE VIEW] |
| URL Base de Media | <input type="text" value="https://d23je69vyjsf8o.cloudfront.net/media/"/> | [STORE VIEW] |
| URL Base de JavaScript | <input type="text" value="{{secure_base_url}}js/"/> | [STORE VIEW] |
| | ▲ Warning! When using CDN, in some cases JavaScript may not run properly if CDN is not in your subdomain | |

Ilustración 23: Configuración CDN Magento para conexiones HTTPS

Hemos omitido la configuración del CDN en la generación de las URL de JS debido a que, como indica el aviso, si no se hace uso de un subdominio del dominio “Base” no funciona correctamente. Hemos de tener en cuenta que si hemos activado “Merge JS y CSS”, el archivo css y js resultante se almacena en /media por lo que, en principio, los únicos archivos Js que no se estarían distribuyendo mediante el CDN serían los propios del Backoffice de Magento, lo cual no tiene importancia.

Debemos de recordar que si hacemos uso de un subdominio de nuestra URL Base, las conexiones HTTPS a CloudFront generarán errores al usar dicho sistema por defecto un certificado SSL firmado para el dominio “cloudfront.com”. Por tanto, para poder hacer uso de un subdominio deberíamos subir a CloudFront nuestro propio certificado SSL válido para nuestro dominio/subdominio.

En el navegador podremos comprobar que nuestro CDN funciona correctamente:

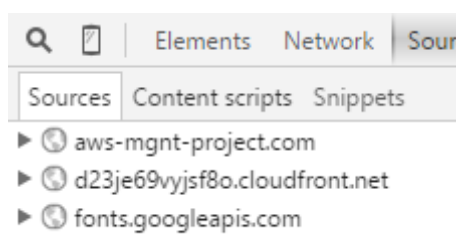


Ilustración 24: CDN en herramientas para desarrollador - 1

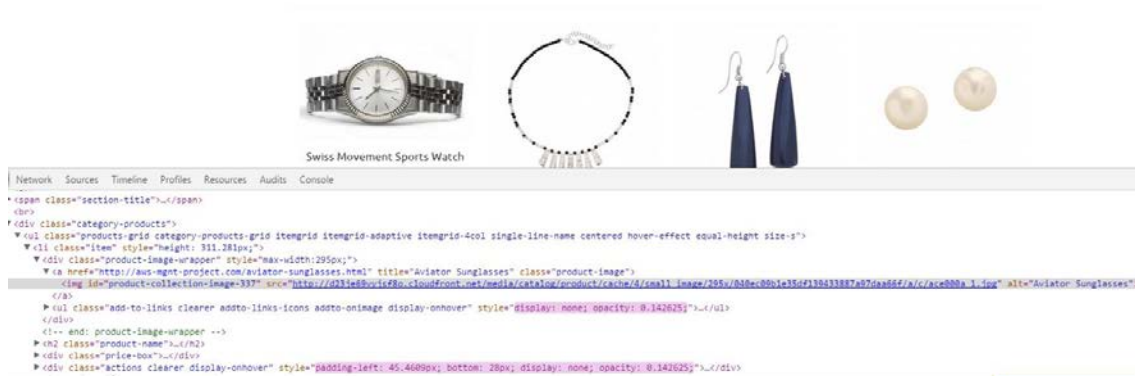


Ilustración 25: CDN en herramientas para desarrollador - 2

4.3.2.5 Deployment

Las herramientas de “Deployment” o despliegue nos permiten automatizar procesos de actualización o modificación de aplicaciones en múltiples servidores. En nuestro caso concreto, en el que haremos uso de imágenes de un sistema autoescalable que hace uso de una imagen AMI para levantar nuevas máquinas, es importante tener algún sistema que permita el despliegue de código de forma automática tanto para las máquinas en funcionamiento como las nuevas que se generen.

De esta forma, no deberemos actualizar la imagen base (lo que implicaría paralizar el sistema para configurar un nuevo Grupo de escalado) en el momento que modifiquemos el código de nuestro Magento, sea por realizar pequeñas modificaciones o por que precisamos de la instalación de algún módulo o similar.

El primer paso sería la configuración de la herramienta de despliegue con las modificaciones a introducir. Para que las modificaciones sean instaladas, las instancias en funcionamiento deben realizar una conexión periódica al servidor de despliegue y las instancias de nueva creación no deben estar disponibles hasta que se hayan conectado al servidor de despliegue y hayan introducido las modificaciones.

Existen numerosas herramientas de despliegue tales como Cheff Puppet, Capistrano [150] o CodeDeploy [151], siendo éste último un servicio dentro de AWS. En nuestro caso haremos uso de CodeDeploy que, a las ventajas explicadas anteriormente, se añade el hecho de la facilidad de conexión de CodeDeploy con las instancias EC2 al ser parte de AWS.

4.3.2.6 Configuración final

Una vez realizadas todas las configuraciones necesarias en Magento, debemos habilitar algunas características en el servidor Web de cara a generar un mejor rendimiento, tales como habilitar la compresión Gzip [153] en todo el contenido enviado y añadir Cabeceras de tiempo de vida.

4.3.2.6.1 Gzip

Al habilitar la compresión de todos los elementos enviados por el servidor web reducimos considerablemente el tamaño del contenido a enviar, con lo que reducimos el tiempo de carga de la web. Lo habilitaremos a través del archivo `.htaccess` ubicado en la raíz de Magento.

En éste archivo, si hemos configurado el servidor web para permitir su uso, podremos incluir configuración específica para el servidor web. Magento por defecto incluye un fichero `.htaccess` ya configurado con todas las opciones, pero muchas de ellas deshabilitadas.

Para permitir la compresión Gzip deberemos asegurarnos de que la siguiente línea no está comentadas (es decir, no tiene una almohadilla al comienzo de la misma):

```
SetOutputFilter DEFLATE
```

A partir de este momento, todo el contenido enviado por el servidor web estará comprimido.

4.3.2.6.2 Expire Headers

Este tipo de cabeceras permiten indicar un tiempo de vida para cada elemento servido por el servidor web. De esta forma, mientras dicho tiempo no haya expirado, no volverá a solicitar ese recurso al servidor, ahorrando tiempo de carga de la web y reduciendo la carga de servidor.

Al igual que para la activación de la compresión, para insertar un el tiempo de vida lo haremos modificando una línea en el fichero `.htaccess`. En este caso, debemos comprobar que el siguiente conjunto de líneas no está comentado:

```
ExpiresActive On  
ExpiresDefault "access plus 1 week"
```

Podemos modificar el tiempo de vida de la forma que más nos convenga.

Una vez que hemos finalizado todas estas configuraciones, deberemos realizar un “Snapshot” o imagen de nuestro nodo web, de forma que podamos configurarlo como imagen base a utilizar para el grupo de Autoescalado, como ya explicamos anteriormente.

4.3.4 Pruebas y validación

4.3.4.1 Pruebas

Para proceder a realizar los test hemos hecho uso de una herramienta online específica para realizar test de estrés a Magento <http://www.magespeedtest.com/>

Los datos de uso de CPU proporcionados por top [154] o htop [155] en entornos virtualizados no tienen por qué corresponderse con la realidad y pueden ser muy diferentes a los proporcionados por CloudWatch.

Para cada prueba se realiza el mismo test: 40 usuarios simultáneos durante 120 segundos. La herramienta se basa en el Sitemap [156] de la web, por lo que el test es lo más real posible al recorrer todas las páginas y no sólo recargar la home.

Los test se han realizado de forma consecutiva, por lo que hay que tener en cuenta que en las gráficas de algunos test (sobre todo las gráficas de RDS) aparecerán los datos de las pruebas anteriores.

4.3.4.1.1 Prueba 1

Para esta primera prueba hemos partido del diagrama final elegido (Ilustración 18 – Diagrama modificado Sistema Autoescable y tolerante a fallos en AWS) pero dejando desactivado el autoescalado, de forma que dos servidores webs serán los encargados de gestionar las peticiones de clientes.

Además, hemos desactivado todos los sistemas de cacheado de Magento por lo que, para cada petición, Magento tendrá que leer e interpretar todos los archivos implicados en el “dibujo” de cada página (archivos php del core o módulo, archivos phtml de la plantilla, base de datos, etc) lo cual ralentiza la web y hace un uso intensivo tanto de la CPU/memoria como la de la base de datos.

Tampoco hemos activado el servicio CDN, por lo que cada una de las imágenes así como los archivos css y js minificados [157] se descargarán de S3 (a través del sistema S3FS).

Los resultados del test nos dejan la siguiente información:

```

**
SIEGE 3.1.0
** Preparing 40 concurrent users for battle.
The server is now under siege... done.
siege aborted due to excessive socket
failure; you
can change the failure threshold in
$HOME/.siegerc

Transactions:                277 hits
Availability:                20.67 %
Elapsed time:                70.00 secs
Data transferred:           9.70 MB
Response time:               7.64 secs
Transaction rate:            3.96
trans/sec
Throughput:                  0.14 MB/sec
Concurrency:                 30.23
Successful transactions:      161
Failed transactions:          1063
Longest transaction:          12.27
Shortest transaction:          0.00

```

Ilustración 26: Log Test sin ninguna caché activada

Como podemos ver la disponibilidad de nuestra tienda con esta configuración ha sido del 20,67%, es decir, se han perdido prácticamente un 80% de las peticiones a nuestra web.

Como no teníamos activado ningún sistema de cacheo, los dos servidores web se han visto superados por la carga de trabajo. Cuando el balanceador web detecta esta situación, empieza a encolar las conexiones para reducir la carga en los servidores y que puedan atender las conexiones que ya tienen establecidas. A pesar de ello, muchas de las conexiones encoladas han expirado su tiempo de vida y se han perdido.

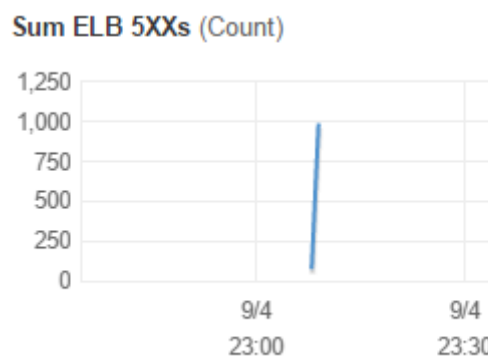


Ilustración 27: Aumento de errores 500 ELB por tiempo de sesión expirado. Prueba 1

Ello ha provocado que la herramienta de testeo haya finalizado antes de lo programado, dado el porcentaje tan alto de conexiones fallidas.

En la siguiente imagen podemos ver una captura de htop de los dos servidores web donde se puede apreciar la alta carga de la CPU:

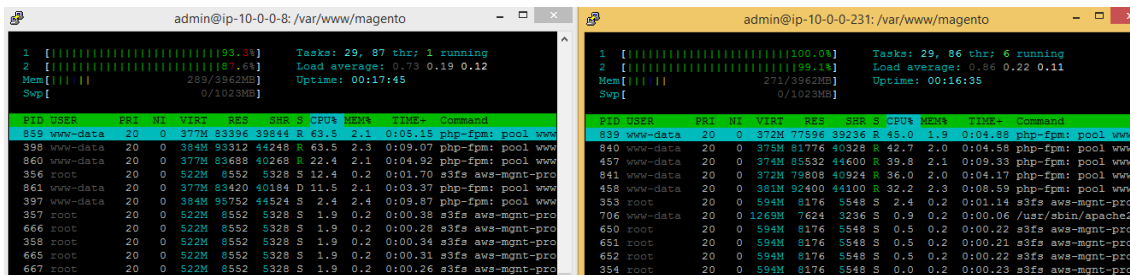


Ilustración 28: Carga servidor Htop. Prueba 1

En cambio, si lo comparamos con los datos de CPU obtenidos por las métricas de CloudWatch podemos ver que el porcentaje de carga es inferior:

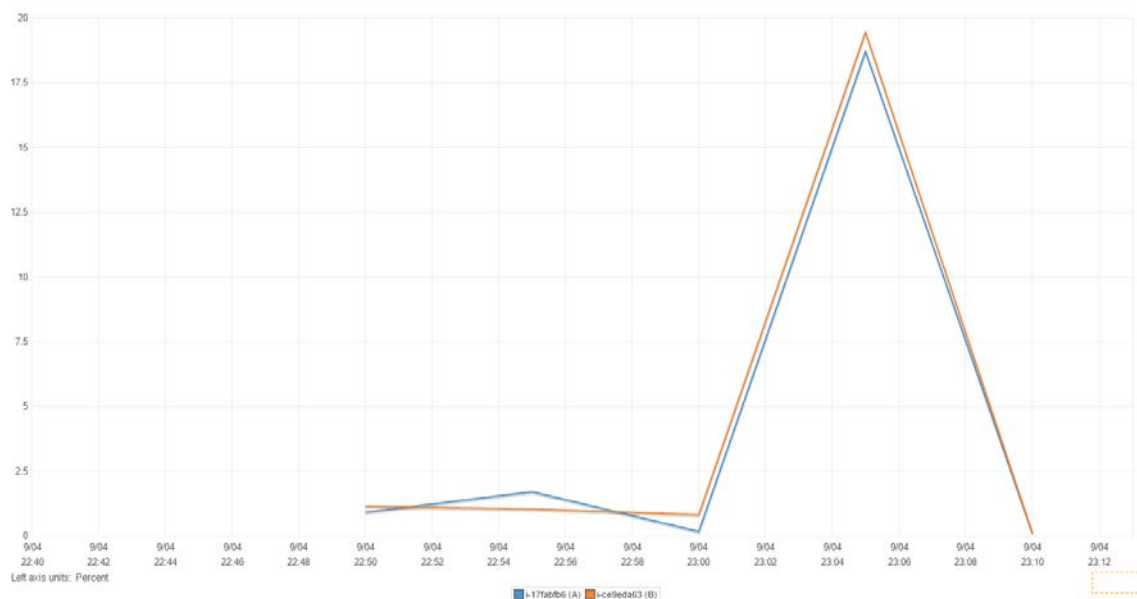


Ilustración 29: Gráfico de CloudWatch del porcentaje de carga de las CPU de los dos servidores web. Prueba 1

Las explicaciones a estas diferencias son varias:

Por un lado los valores de htop pueden estar contaminados al estar en un sistema virtualizado, ya que los valores de CPU que obtiene htop vienen datos por el Hipervisor [158], con lo que es posible que se estén sumando valores de más instancias.

Por otro lado, en el caso de la gráfica de CloudWatch los valores representados son una media. Si tenemos en cuenta que el balanceador empezó a encolar peticiones y que al final se acabó cortando la conexión, es probable que no sólo las instancias se saturasen, sino también al final el propio balanceador. Probablemente en los momentos previos a interrumpirse el test la carga de la CPU fuese muy pequeña al estar haber acabado de resolver peticiones y no recibir nuevas por estar el balanceador colapsado.

No debemos olvidar que el balanceador es un servicio del cual no elegimos su capacidad, si no que autoescala en función de la demanda. Como en el resto de servicios, el autoescalado no es instantáneo, por lo que si el aumento de carga es muy repentino, puede haber momentos de colapso.

A todo esto hay que añadir que las instancias t2 de Amazon están concebidas para un uso en “ráfagas” de la CPU por lo que, dependiendo del tipo de instancia, sus valores normales de uso deberían estar en torno al 20% pero, si fuese necesario, se podría llegar a valores por encima del 100% durante un tiempo limitado (controlado por los créditos de uso de CPU disponibles).

Con esta configuración claramente nuestro sistema de comercio electrónico no es capaz de gestionar un aumento de demanda.

Vamos a analizar por encima otras métricas:

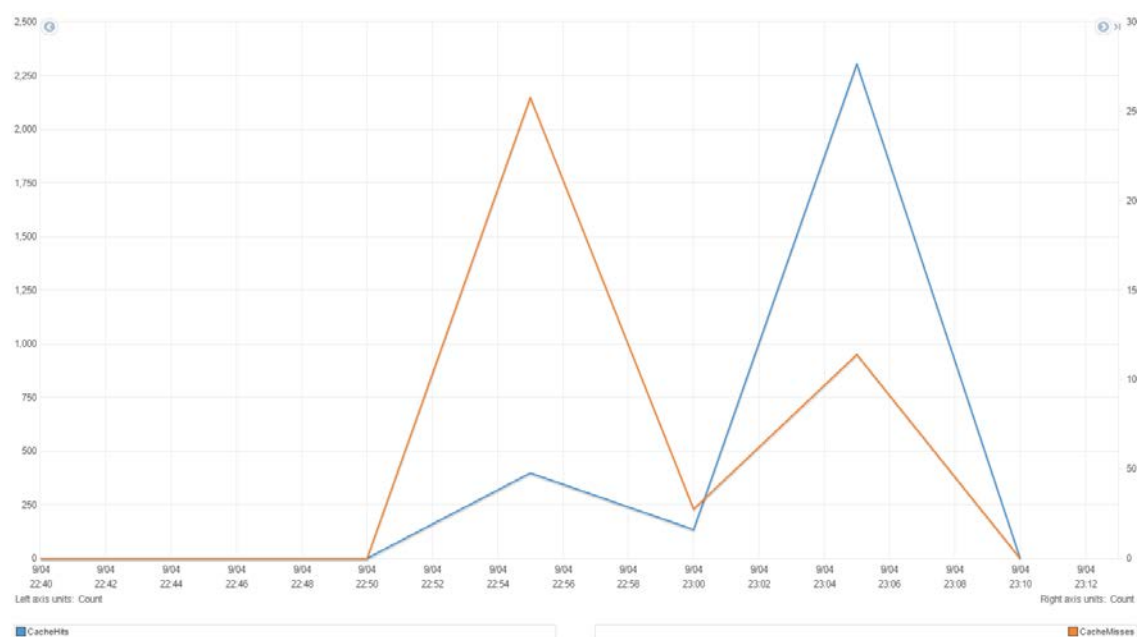


Ilustración 30: Métrica CloudWatch del servidor Redis. Prueba 1

En los datos de gráfico anterior podemos ver cómo, al estar la caché desactivada, no recibe prácticamente peticiones. A pesar de ello, Magento siempre mantiene un pequeño uso de la caché, de ahí que sí que se genere tráfico. Se puede ver cómo en un primer momento (donde se hace un uso normal de la web) hay los mismos fallos de caché que aciertos pero, minutos después, cuando se ejecuta el test de stress, se obtiene un porcentaje mayor de acierto al estar ya los pocos datos solicitados, cacheados.

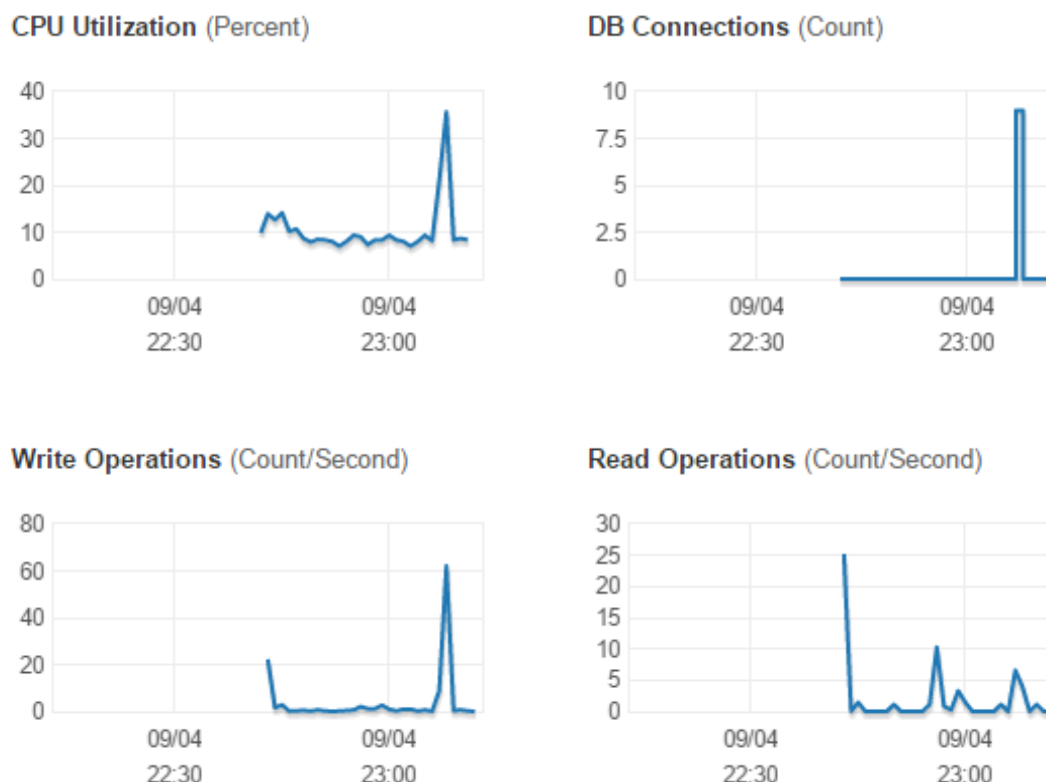


Ilustración 31: Métricas RDS (base de datos). Prueba 1

En las métricas obtenidas de la base de datos podemos observar el momento en el que se inicia el test de stress pasadas las 23h. La CPU recibe un pico de carga al igual que el número de conexiones a la base de datos. Estos “picos” tienen una corta duración debido a la rápida finalización del test por la baja disponibilidad del sistema.

Tanto en las gráficas de escritura como lectura de datos podemos ver actividad anterior a la hora de inicio del test. Esto se debe a la preparación que se hizo del sistema para su puesta en marcha. Sobre todo en el caso de la gráfica de lectura podemos ver un pico significativo provocado por el re-indexado de tablas de la base de datos.

Una vez iniciado el test (sobre las 23:05h), se puede ver claramente un aumento significativo de la escritura de datos y un pequeño pico de lectura. Para entender esto debemos conocer que Magento realiza un “logueo” exhaustivo del comportamiento de los usuarios con el fin de poder realizar todo tipo de estadísticas relativas a usuarios y a productos. Dicha información se almacena en base de datos, lo que implica una alta necesidad de escritura.

Para finalizar con esta prueba, realizamos dos test de rendimiento haciendo uso de dos herramientas diferentes. Una de ella es Gtmetrix [159], con sus servidores centrales ubicados en Canadá. La otra es WebPageTest [160], la cual nos permite seleccionar desde dónde queremos realizar la petición a nuestro servidor, por lo que hemos elegido París. Esto va a suponer que es más que probable que los tiempos de carga sean diferentes, entre otras cosas, por la diferente localización.

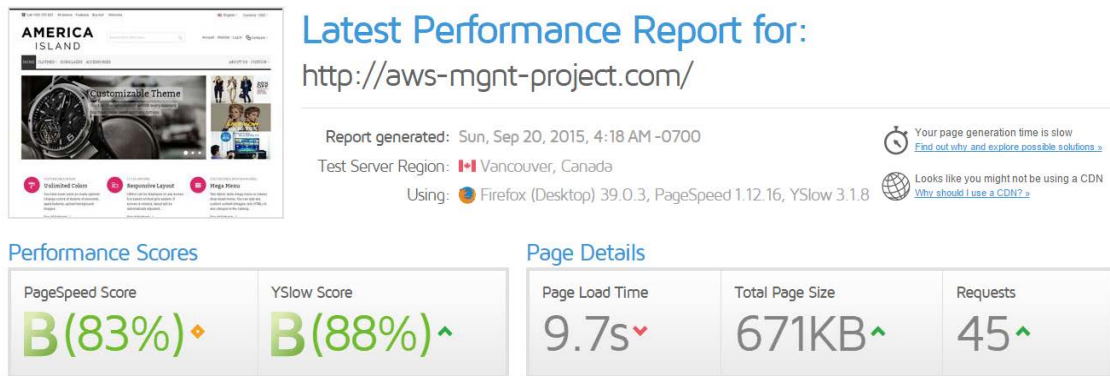


Ilustración 32: Test rendimiento Gtmetrix.com. Prueba 1

En el test realizado con la herramienta web “Gtmetrix” podemos ver cómo obtenemos un tiempo de acceso muy elevado (9,7 segundos).

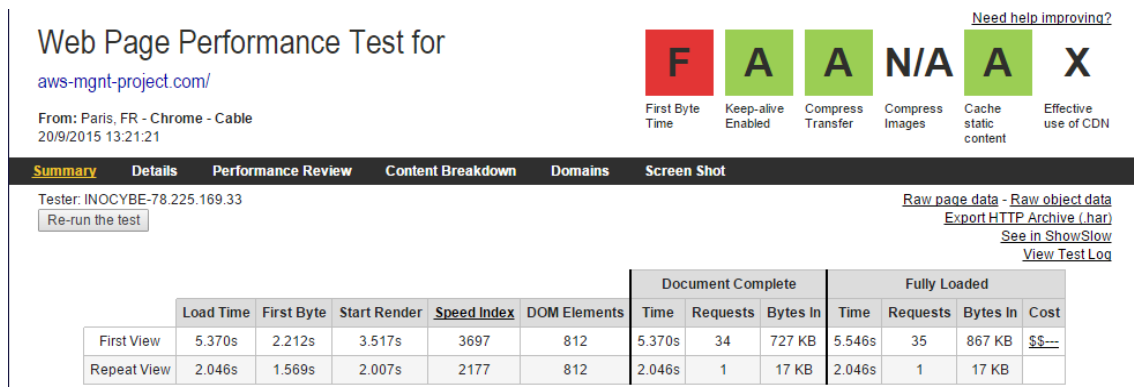


Ilustración 33: Test rendimiento WebPageTest.com. Prueba 1

En el test realizado con la herramienta web “Webpagetest” podemos ver cómo obtenemos un tiempo de acceso elevado (5,3 segundos) pero menor que en la herramienta anterior.

Un factor que influye en las diferencias de tiempo en entre las herramientas, es el hecho de que al estar ubicadas en países que pertenecen a diferentes Websites, cargan tiendas diferentes, con lo que puede haber diferencias en el número de banners, productos, etc que deben cargar.

En cualquier caso, ambas herramientas obtienen tiempos muy elevados, a los que contribuye el hecho de que las imágenes y los archivos css y js estén siendo descargados desde S3 a través de S3FS, como podemos ver en la siguiente imagen:

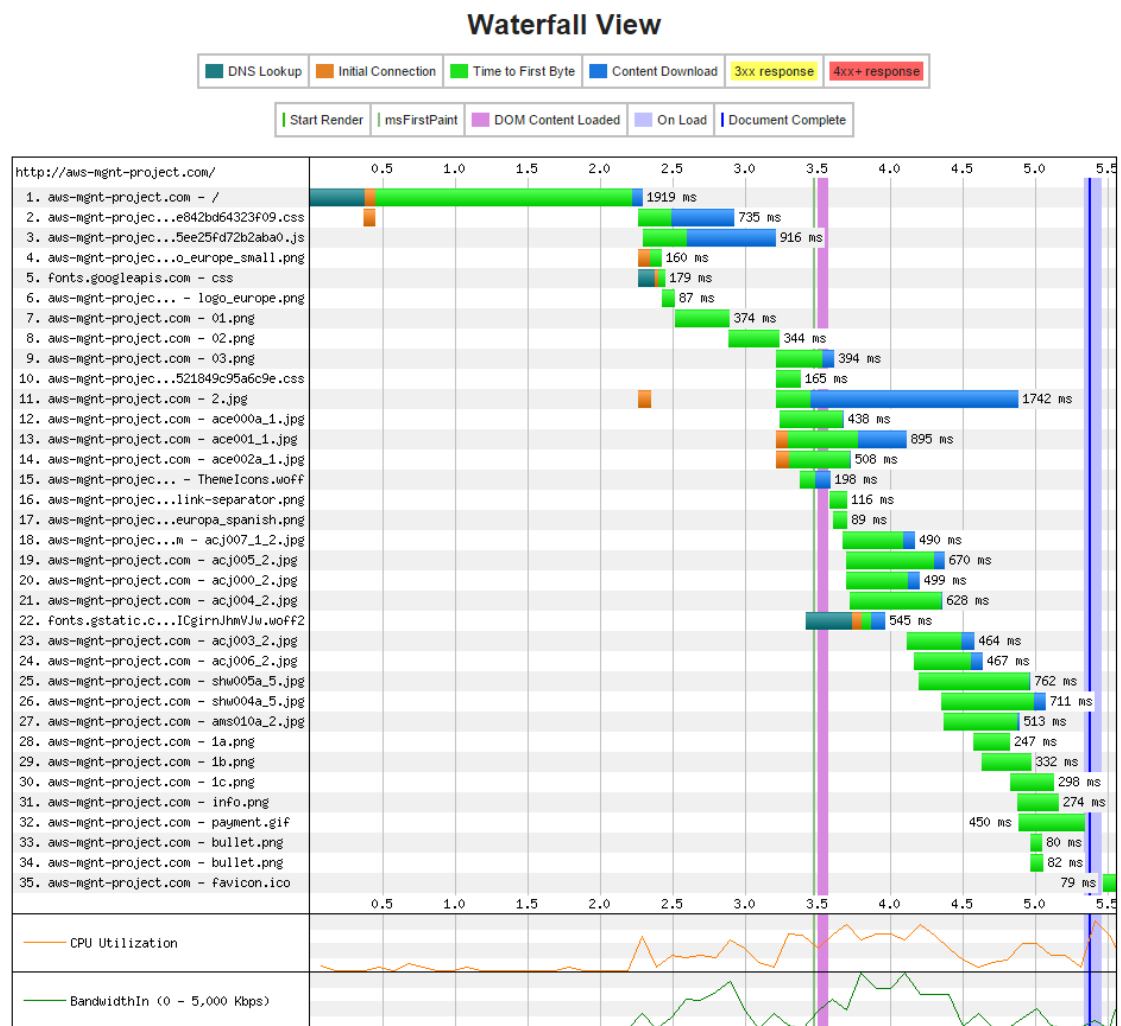


Ilustración 34: Waterfall WebPageTest.com. Prueba 1

Las imágenes tienen un tiempo de descarga altísimo. El servicio S3 no está pensado para proporcionar un alto rendimiento. Si además se accede a los datos en S3 a través de un sistema que hace de intermediario (S3FS), las latencias acumuladas suponen un gran impacto en el rendimiento de la web.

4.3.4.1.2 Prueba 2

Para esta prueba partimos del mismo sistema y configuración que para la Prueba 1, pero en este caso hemos habilitado la caché de Magento la cual, recordemos, almacenará los datos en un servidor Redis.

```

SIEGE 3.1.0
** Preparing 40 concurrent users for battle.
The server is now under siege...
Lifting the server siege...-      done.

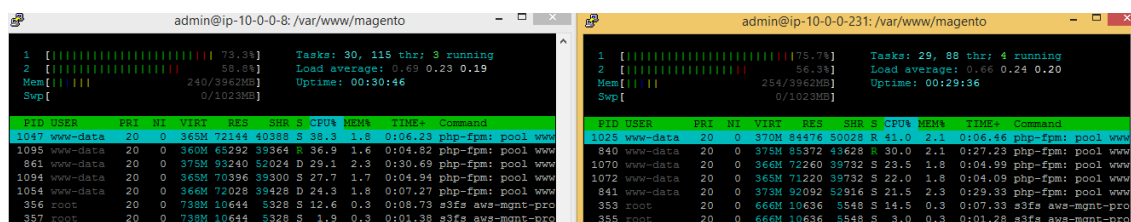
Transactions:          1588 hits
Availability:          100.00 %
Elapsed time:          119.39 secs
Data transferred:      54.09 MB
Response time:          2.48 secs
Transaction rate:      13.30
trans/sec
Throughput:             0.45 MB/sec
Concurrency:           32.95
Successful transactions: 972
Failed transactions:    0
Longest transaction:    6.38
Shortest transaction:   0.26

```

Ilustración 35: Log Test con la caché de Magento activada. Prueba 2

Podemos observar cómo gracias a tener habilitada la caché, en esta prueba hemos obtenido un 100% de disponibilidad además de un nivel de concurrencia mayor, transacciones de menor duración, etc. Es decir, mejores números en general.

Al evaluar las mismas métricas que en el apartado anterior, podemos ver una mejora general de los datos. En el caso de las gráficas, debemos tener en cuenta que este segundo test comienza a las 23:15h, con lo que debemos ignorar datos anteriores (pertenecientes a la anterior prueba).



| PID | USER | PR | NI | VI | RES | SR | S | CPU% | MEM% | TIME | Command |
|------|----------|----|----|------|-------|-------|---|------|------|---------|-------------------|
| 1067 | www-data | 20 | 0 | 365M | 72144 | 40388 | S | 38.3 | 1.8 | 0:06.23 | php-fpm: pool www |
| 1095 | www-data | 20 | 0 | 360M | 65292 | 39364 | R | 36.9 | 1.6 | 0:04.82 | php-fpm: pool www |
| 861 | www-data | 20 | 0 | 375M | 93240 | 52024 | D | 29.1 | 2.3 | 0:30.69 | php-fpm: pool www |
| 1094 | www-data | 20 | 0 | 365M | 70396 | 39300 | S | 27.7 | 1.7 | 0:04.94 | php-fpm: pool www |
| 1054 | www-data | 20 | 0 | 366M | 72028 | 39428 | D | 24.3 | 1.8 | 0:07.27 | php-fpm: pool www |
| 356 | root | 20 | 0 | 738M | 10644 | 5328 | S | 12.6 | 0.3 | 0:08.79 | s3fs aws-mgmt-pro |
| 357 | root | 20 | 0 | 738M | 10644 | 5328 | S | 1.9 | 0.3 | 0:01.38 | s3fs aws-mgmt-pro |

Ilustración 36: Carga servidor Htop. Prueba 2

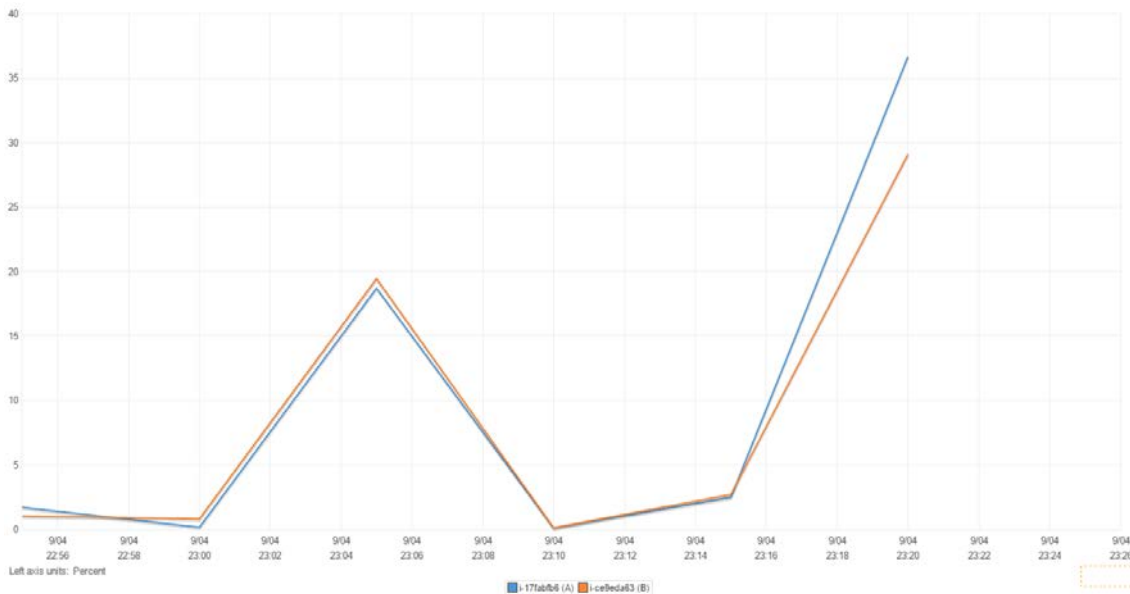


Ilustración 37: Gráfico de CloudWatch del porcentaje de carga de las CPU de los servidores web. Prueba 2

Como en la prueba anterior, las dos gráficas mostradas nos indican los valores de uso de la CPU medidas con distintas herramientas.

En el caso de la medición directa en el servidor con `htop`, aunque debemos recordar que no es fiable ya que puede estar contaminada con datos de otros servidores, observamos una disminución de la carga en los dos servidores.

En el caso de la gráfica de CloudWatch (Ilustración 37), observamos que el pico de uso es mucho mayor que para la anterior prueba. Debemos recordar que esta gráfica muestra una media de uso y que en el caso anterior el test se cortó, por lo que los resultados son lógicos. En cualquier caso, el uso de CPU no sobrepasa el 40% de media.

Si analizamos las métricas del servidor Redis, veremos una diferencia significativa con respecto a la anterior prueba:

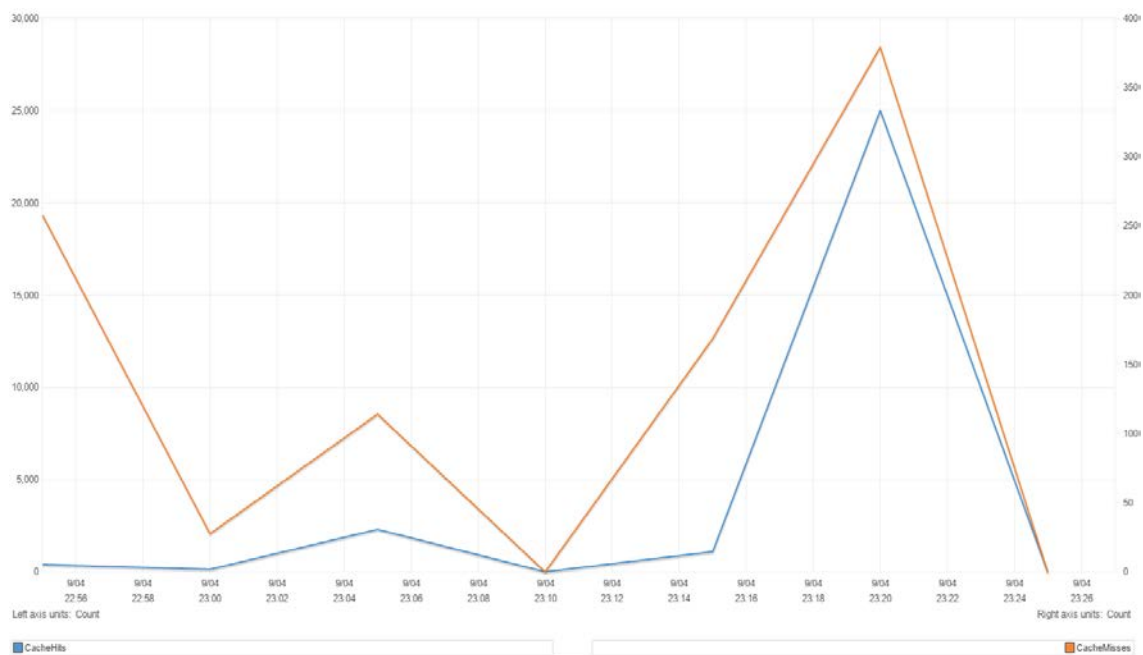


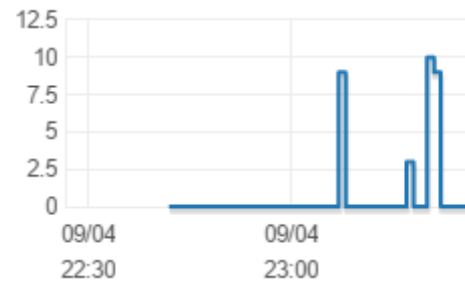
Ilustración 38: Métrica CloudWatch del servidor Redis. Prueba 2

Observamos como los accesos a caché han aumentado enormemente ahora que hemos activado la caché. Es normal un repunte de los fallos de caché ya que el primer acceso a una página va a fallar y Magento tiene que generar los datos, pero a partir de ese momento será el servidor Redis el que sirva dicha página, pasando a ser un “hit” y descargando Magento (y por tanto, el servidor web y la base de datos). Se puede ver cómo aunque los fallos de caché se han multiplicado por cuatro, los aciertos lo han hecho por más de diez.

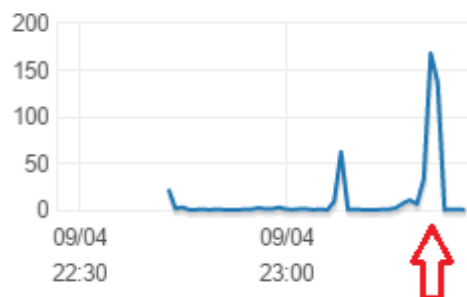
CPU Utilization (Percent)



DB Connections (Count)



Write Operations (Count/Second)



Read Operations (Count/Second)

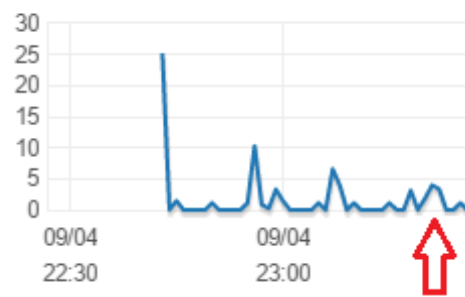


Ilustración 39: Métricas RDS (base de datos). Prueba 2

En el gráfico con las métricas de la base de datos podemos ver cómo, con respecto a la prueba 1, se incrementa el uso de la CPU y de la escritura en disco, pero se reduce la lectura de disco. Se han señalado con flechas rojas la actividad correspondiente a esta prueba.

El incremento en el uso de la CPU está claramente relacionado con el incremento en las operaciones de escritura en casi el doble con respecto al a prueba 1. Al ser ya capaces los servidores web de atender todas las peticiones, lógicamente se ven incrementadas las operaciones de escritura por el constante y metódico registro de la actividad de los usuarios en base de datos por parte de Magento. Por el contrario, gracias a la actuación de la caché, a pesar de multiplicarse por cuatro hasta llegar al 100% la disponibilidad de la tienda, incrementarse en cinco las transferencias completadas o el tráfico de datos generado y pese a la intensa actividad en escritura debido a la actividad de los usuarios, las operaciones de escritura no sólo no han aumentado si no que se han reducido, ya que, una vez se han obtenido los datos de la base de datos para “dibujar” una página, ésta se almacena en caché y no vuelve a ser “dibujada” (hasta que el tiempo de vida de la página en caché expire).

Tanto en éste como en subsiguientes test, el balanceador no vuelve a presentar problemas. Lo podemos achacar a la mejora de la disponibilidad de los servidores web, pero también debemos tener en cuenta que, con el primer test, lo hemos “precalentado”. Una de las técnicas para tener el sistema lo más preparado posible antes una avalancha de peticiones, es “atacarlo” con

peticiones generadas por herramientas de stress para que los sistemas como ELB tengan tiempo de prepararse. Es conocida como técnica de “warm up”.

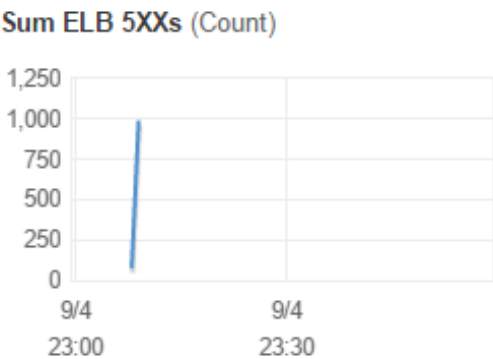


Ilustración 40: Gráfica ELB sin más errores 500 por saturación. Prueba 2.

En nuestro caso, la primera prueba también ha servido al ELB como precalentamiento.

Como en la prueba anterior, para finalizar realizamos dos test de rendimiento haciendo uso de Gtmetrix, y Webpagetest:

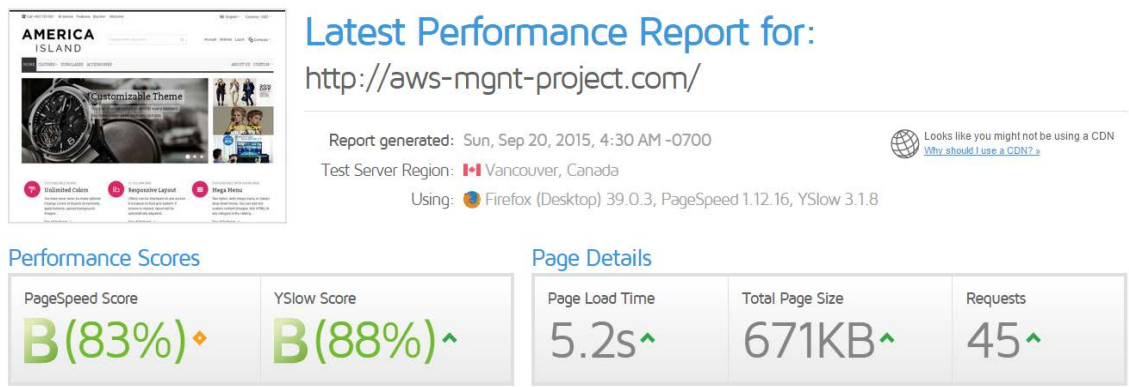


Ilustración 41: Test rendimiento Gtmetrix.com. Prueba 2

En el test realizado con Gtmetrix vemos cómo la activación de la caché a reducido prácticamente a la mitad el tiempo de carga de la web.

Web Page Performance Test for

aws-mgnt-project.com/

From: Paris, FR - Chrome - Cable
20/9/2015 13:31:40

F

First Byte
Time

A

Keep-alive
Enabled

A

Compress
Transfer

N/A

Compress
Images

A

Cache
static
content

X

Effective
use of CDN

[Need help improving?](#)

[Summary](#)

[Details](#)

[Performance Review](#)

[Content Breakdown](#)

[Domains](#)

[Screen Shot](#)

Tester: INOCYBE-78.225.169.33

[Re-run the test](#)

[Raw page data](#) - [Raw object data](#)

[Export HTTP Archive \(.har\)](#)

[View Test Log](#)

| | Load Time | First Byte | Start Render | Speed Index | DOM Elements | Document Complete | | | Fully Loaded | | | |
|-------------|-----------|------------|--------------|-------------|--------------|-------------------|----------|----------|--------------|----------|----------|----------|
| | | | | | | Time | Requests | Bytes In | Time | Requests | Bytes In | Cost |
| First View | 4.435s | 1.384s | 2.509s | 2689 | 812 | 4.435s | 34 | 719 KB | 4.582s | 35 | 867 KB | \$\$.--- |
| Repeat View | 1.780s | 1.305s | 1.706s | 1960 | 812 | 1.780s | 1 | 17 KB | 1.780s | 1 | 17 KB | |

Ilustración 42: Test rendimiento WebPageTest.com. Prueba 2

En el test realizado con WebPageTest vemos cómo la activación de la caché a reducido en casi un segundo el tiempo de carga de la web. El tiempo de carga de las subsiguientes visualizaciones (si solicitáramos el mismo contenido) también se reduce.

Al igual que en la anterior prueba, hemos dejado el servicio de CDN desactivado.

4.3.4.1.3 Prueba 3

En ésta última prueba hemos procedido a activar el uso de “Flat resources”, de los que ya hablamos anteriormente en la sección de optimización para los productos y las categorías. Al ser una mejora específica para la base de datos, es posible que no notemos una gran variación de los parámetros que estamos analizando, a excepción de la base de datos.

```

**

SIEGE 3.1.0
** Preparing 40 concurrent users for battle.
The server is now under siege...
Lifting the server siege...- done.

Transactions:          1561 hits
Availability:          100.00 %
Elapsed time:          119.66 secs
Data transferred:      51.20 MB
Response time:          2.50 secs
Transaction rate:       13.05
trans/sec
Throughput:             0.43 MB/sec
Concurrency:            32.61
Successful transactions: 1017
Failed transactions:     0
Longest transaction:    6.47
Shortest transaction:   0.27
```

Ilustración 43: Log Test con la caché de Magento activada. Prueba 3

Observamos que no hay grandes diferencias con respecto a la anterior prueba.

admin@ip-10-0-0-8: /var/www/magento

1 [|||||] 71.6%

Tasks: 30, 115 thr: 7 running

2 [|||||] 57.1%

Load average: 1.33 0.38 0.21

Mem[|||||] 253/3962MB

Uptime: 01:02:50

Swp[|] 0/1023MB

| PID | USER | PRI | NI | VIRT | RES | SHR | S | CPU% | MEM% | TIME+ | Command |
|------|----------|-----|----|------|-------|-------|---|------|------|---------|-------------------|
| 1095 | www-data | 20 | 0 | 377M | 94308 | 51512 | R | 43.7 | 2.3 | 0:52.07 | php-fpm: pool www |
| 1548 | www-data | 20 | 0 | 365M | 71632 | 38576 | R | 28.6 | 1.8 | 0:15.30 | php-fpm: pool www |
| 1428 | www-data | 20 | 0 | 363M | 69704 | 40068 | D | 27.7 | 1.7 | 0:18.94 | php-fpm: pool www |
| 1414 | www-data | 20 | 0 | 366M | 71872 | 39748 | S | 25.2 | 1.8 | 0:21.37 | php-fpm: pool www |
| 1545 | www-data | 20 | 0 | 365M | 71528 | 40004 | S | 23.8 | 1.8 | 0:18.80 | php-fpm: pool www |
| 356 | root | 20 | 0 | 883M | 11104 | 5328 | S | 17.5 | 0.3 | 0:34.45 | s3fs aws-mgmt-pro |
| 357 | root | 20 | 0 | 883M | 11104 | 5328 | S | 2.4 | 0.3 | 0:04.61 | s3fs aws-mgmt-pro |
| 666 | root | 20 | 0 | 883M | 11104 | 5328 | S | 2.4 | 0.3 | 0:04.45 | s3fs aws-mgmt-pro |
| 862 | root | 20 | 0 | 883M | 11104 | 5328 | S | 1.9 | 0.3 | 0:04.24 | s3fs aws-mgmt-pro |

admin@ip-10-0-0-231: /var/www/magento

1 [|||||] 64.9%

Tasks: 30, 115 thr: 4 running

2 [|||||] 49.3%

Load average: 1.11 0.36 0.21

Mem[|||||] 253/3962MB

Uptime: 01:01:38

Swp[|] 0/1023MB

| PID | USER | PRI | NI | VIRT | RES | SHR | S | CPU% | MEM% | TIME+ | Command |
|------|----------|-----|----|------|-------|-------|---|------|------|---------|-------------------|
| 1550 | www-data | 20 | 0 | 365M | 72088 | 40184 | R | 36.7 | 1.8 | 0:15.76 | php-fpm: pool www |
| 1072 | www-data | 20 | 0 | 365M | 74544 | 43016 | S | 35.7 | 1.8 | 0:52.42 | php-fpm: pool www |
| 1417 | www-data | 20 | 0 | 369M | 85004 | 50964 | R | 21.0 | 2.1 | 0:19.87 | php-fpm: pool www |
| 1549 | www-data | 20 | 0 | 365M | 71944 | 40440 | S | 18.6 | 1.8 | 0:15.25 | php-fpm: pool www |
| 1418 | www-data | 20 | 0 | 372M | 88360 | 51092 | S | 15.2 | 2.2 | 0:17.42 | php-fpm: pool www |
| 353 | root | 20 | 0 | 955M | 11492 | 5548 | S | 11.7 | 0.3 | 0:32.24 | s3fs aws-mgmt-pro |
| 651 | root | 20 | 0 | 955M | 11492 | 5548 | S | 1.5 | 0.3 | 0:04.45 | s3fs aws-mgmt-pro |
| 355 | root | 20 | 0 | 955M | 11492 | 5548 | S | 1.5 | 0.3 | 0:04.38 | s3fs aws-mgmt-pro |
| 652 | root | 20 | 0 | 955M | 11492 | 5548 | R | 1.5 | 0.3 | 0:04.45 | s3fs aws-mgmt-pro |

Ilustración 44: Carga servidor Htop. Prueba 3

Observamos que hay una pequeña disminución de la carga de la CPU, pero al ser una captura en un momento concreto puede ser casualidad.

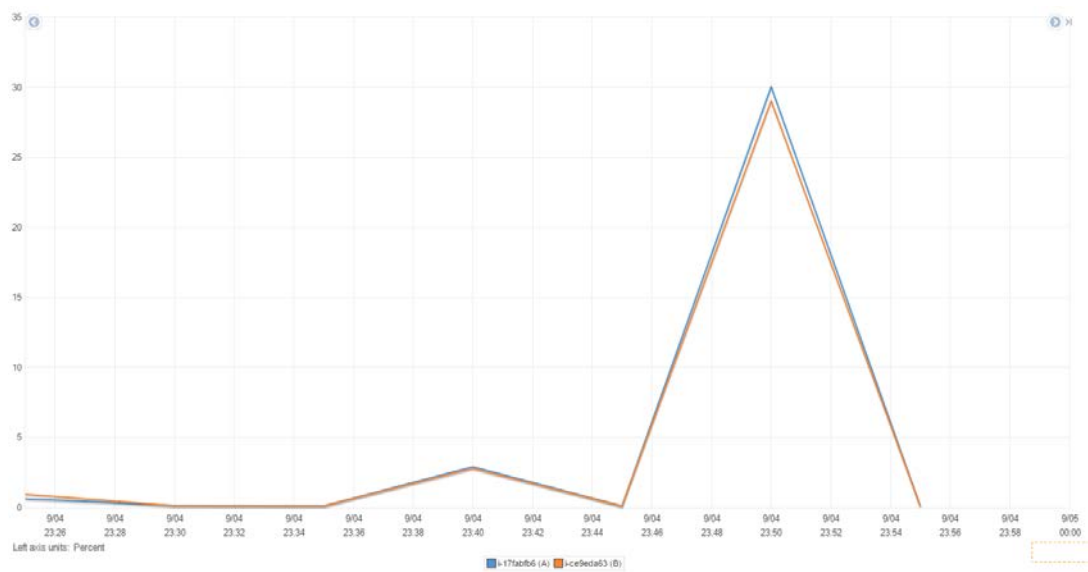


Ilustración 45: Gráfico de CloudWatch del porcentaje de carga de las CPU de los servidores web. Prueba 3

Observamos que, en la línea de lo que nos indicaban los datos de htop, se observa una ligera disminución de la carga de la CPU

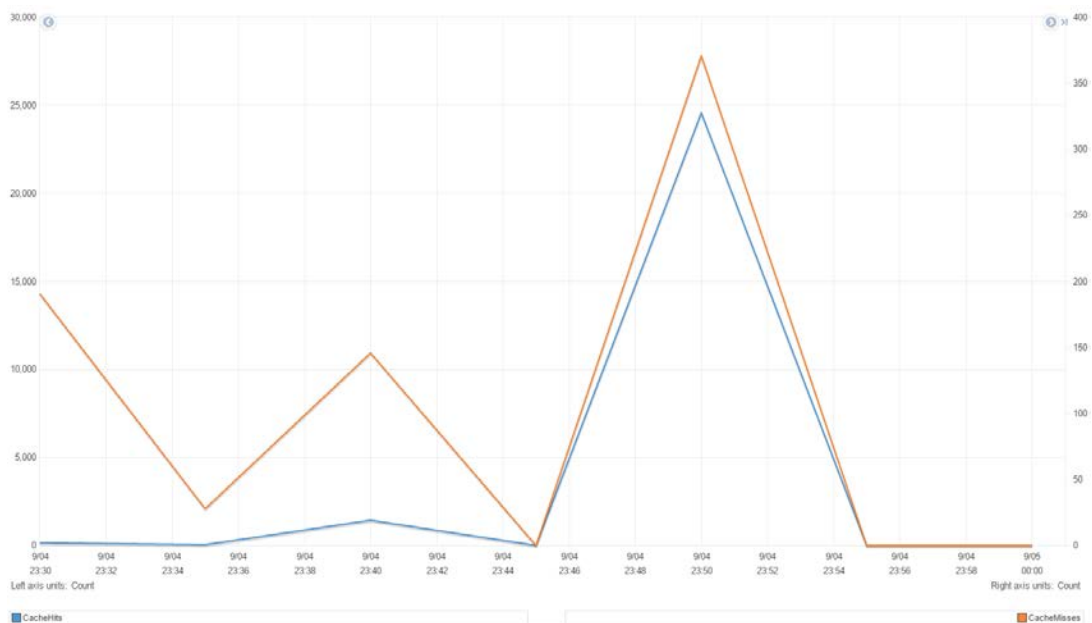


Ilustración 46: Métrica CloudWatch del servidor Redis. Prueba 3

En el caso de las métricas del servidor Redis no se observa ningún tipo de cambio.

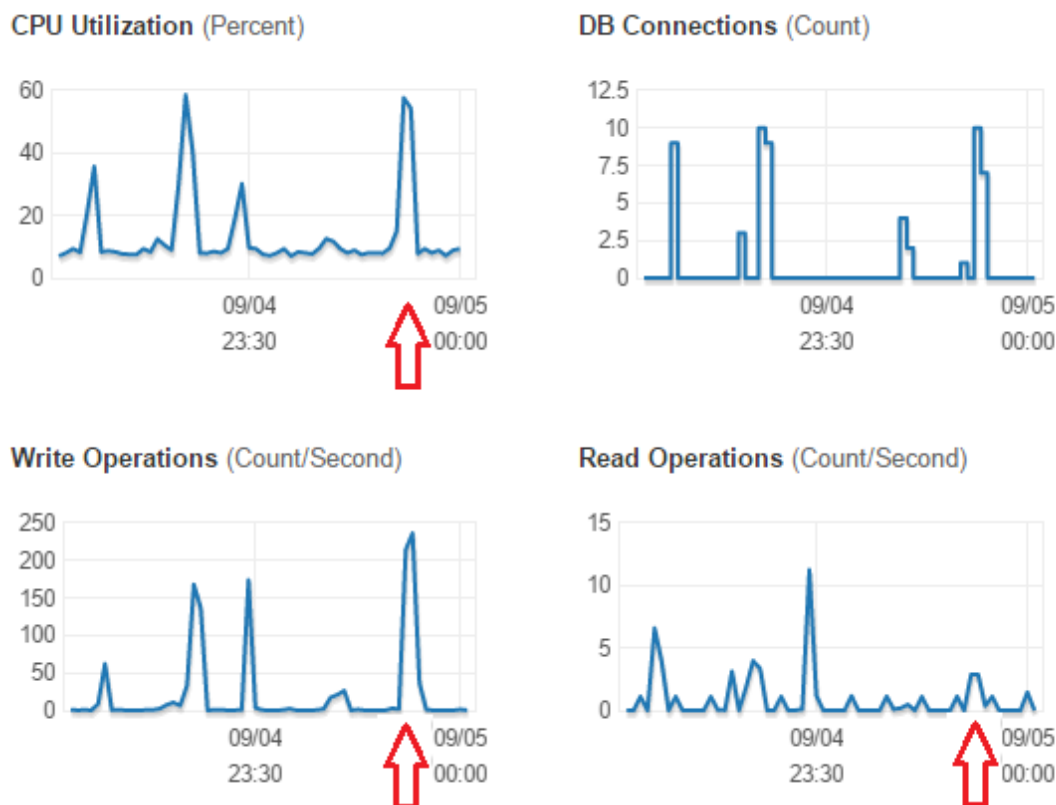


Ilustración 47: Métricas RDS (base de datos). Prueba 3

En las métricas de la base de datos sí que podemos observar cambios destacables después de haber activado las opciones “Flat”. Se ha indicado con flechas rojas los datos correspondientes a ésta última prueba. Para poder realizar la comparación, debemos tener en cuenta que los datos que aparecen reflejados a las 23:30h son generados por el indexado necesario una vez que se han activado las opciones “Flat”. Dicho indexado implica una gran cantidad de operaciones de lectura y escritura.

El primer dato destacable en esta nueva prueba con las opciones “Flat” activadas es el incremento considerable con respecto a la anterior prueba de las operaciones de escritura. Debemos tener en cuenta que, aunque se ha realizado un indexado de productos para rellenar las tablas “Flat”, éste no es completo, por lo que a las operaciones de escritura habituales en esta prueba debemos sumar las que se produzcan con el objetivo de completar la tabla ante peticiones de productos y/o categorías.

El segundo dato a destacar es la reducción, aunque modesta, en las operaciones de lectura. Esto debemos achacarlo al hecho de que, por ejemplo, para obtener todos los datos de un producto la consulta a la base de datos es mucho más sencilla al tener que acceder a una sola tabla.

Aunque en general no se aprecian cambios significativos en las métricas que puedan hacernos pensar en una mejora del rendimiento del sistema (exceptuando la disminución de carga de la CPU y los cambios en RDS), debemos tener en cuenta que la mejora implementada afecta directamente al tiempo de la consulta, lo cual marca la diferencia sobre todo en páginas de catálogo con gran cantidad de productos y en páginas de producto (sobre todo no cacheadas). Si bien es cierto que en el fondo acaba afectando al sistema, en pruebas con métricas tan específicas puede ser complicado verificar dicha mejora.

Como en pruebas anteriores, para finalizar realizamos dos test de rendimiento haciendo uso de Gtmetrix, y Webpagetest. En esto caso, además, hemos activado el uso del CDN que hemos configurado:

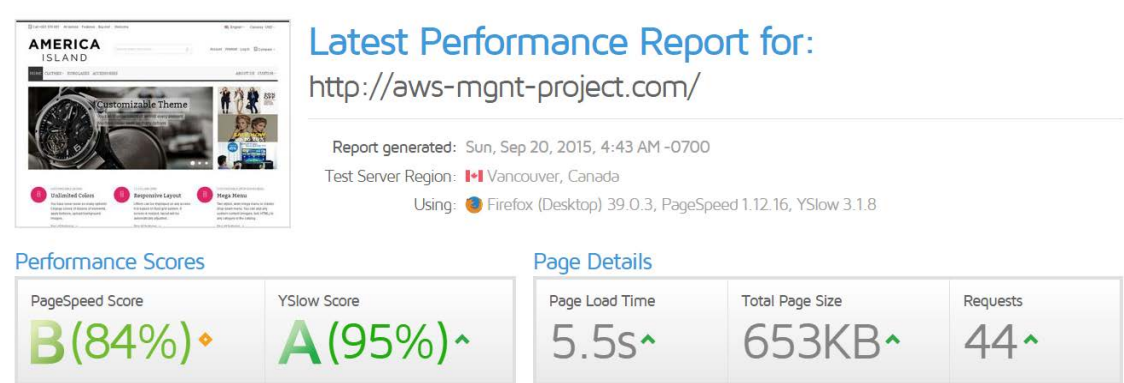


Ilustración 48: Test rendimiento Gtmetrix.com. Prueba 3

Observamos que el test realizado con Gtmetrix no sólo no reduce su tiempo de carga si no que lo aumenta ligeramente. Viendo los tiempos anteriores en comparación con la otra herramienta que hemos estado utilizando, parece claro que el hecho de cargar tiendas distintas y, sobre todo, la distancia al servidor, afecta profundamente a la velocidad de transmisión de los datos y hace que éstos fluctúen.

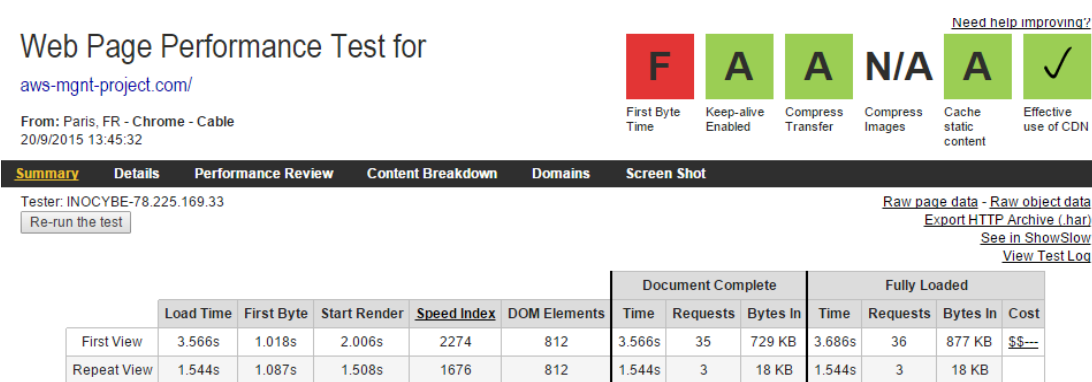


Ilustración 49: Test rendimiento WebPageTest.com. Prueba 3

Como viene ocurriendo en las distintas pruebas, la herramienta WebPageTest vuelve a mostrar una reducción en el tiempo de carga de la web, sea en la primera carga o en sucesivas, obteniendo valores muy positivos si tenemos en cuenta la envergadura del sistema.

Como hemos comentado, para esta prueba hemos activado y configurado el servicio CDN el cual, como veremos en la imagen siguiente, tiene un impacto significativo en el tiempo de carga al proporcionar todos los elementos de una forma mucho más rápida y con menores latencias y, además, descargando a los servidores web de trabajo.

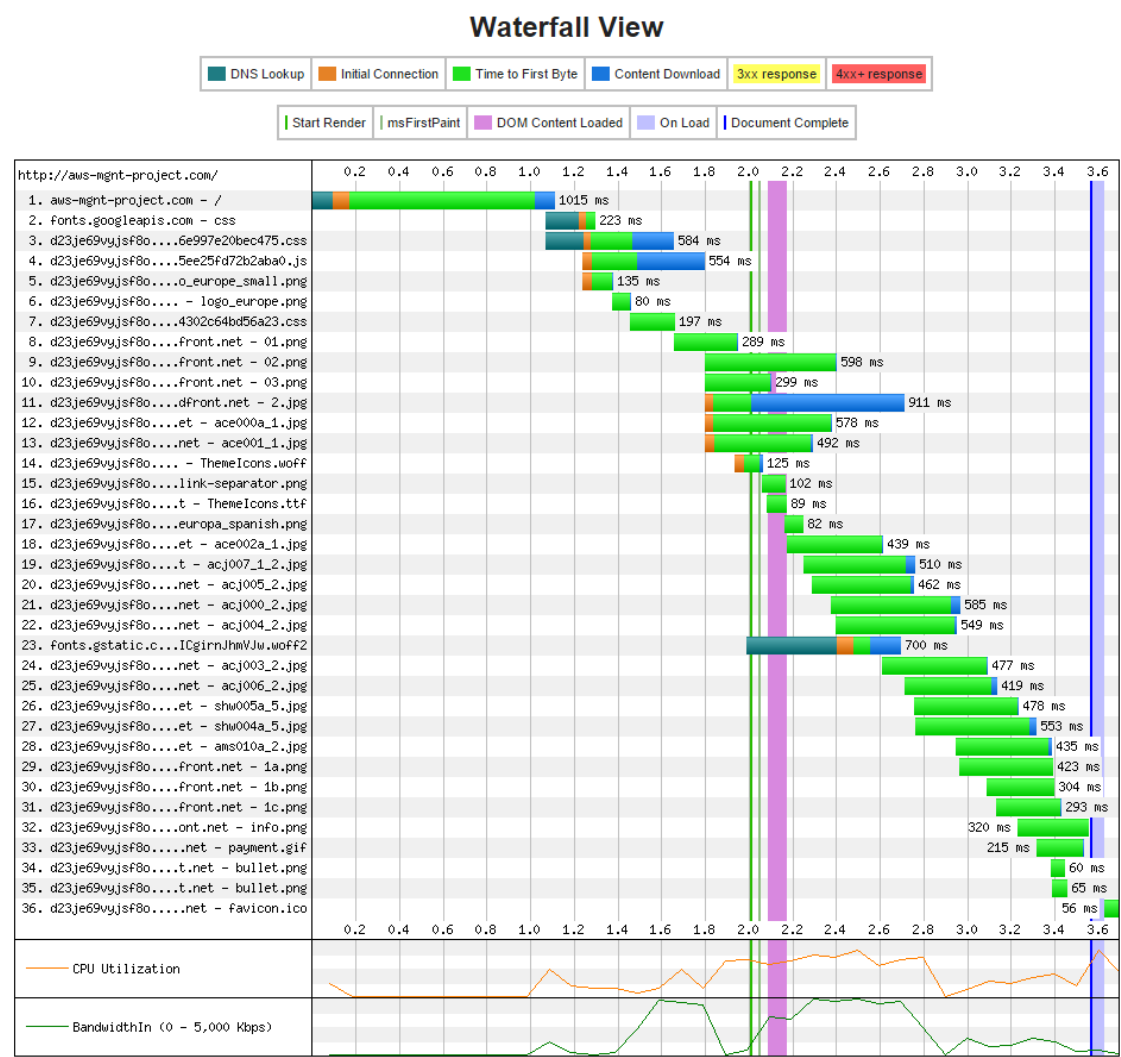


Ilustración 50: Waterfall WebPageTest.com. Prueba 3

4.3.4.1.4 Prueba Autoescalado

4.3.4.1.4.1 Configuración

En esta prueba partimos de la misma configuración de nuestro sistema que en la prueba anterior, pero en un entorno autoescalado.

Para crear dicho entorno se ha configurado un “Grupo de autoescalado” que hace uso de la misma imagen AMI usada en las instancias de pruebas anteriores. Se configura un mínimo de dos instancias (una en cada zona de disponibilidad), un máximo de ocho y se crean las siguientes reglas:

- En caso de que el uso medio de CPU de todas las instancias sea igual o mayor al 60% de forma consecutiva durante al menos un periodo de cinco minutos, se añadirá una instancia.
- En caso de que el uso medio de CPU de todas las instancias sea igual o menor al 30% de forma consecutiva durante al menos un periodo de cinco minutos, se eliminará una instancia.
- Entre cada disparo de una regla, se han de esperar 600 segundos para dar tiempo a que la nueva instancia se cree y esté lista.
- Se establece una ventana de 450 segundos hasta que el balanceador activa los “Healthy checks” para comprobar que la instancia funciona correctamente.

Es importante configurar correctamente todos los parámetros y se deben hacer varias pruebas. Una configuración errónea puede producir desde que los servidores lleguen a colapsarse por que se ha esperado demasiado para iniciar instancias de apoyo, hasta que haya siempre online exceso de recursos disponibles, lo que repercute en el coste. Además, se deben realizar varias pruebas a la hora de dar por configurados los dos últimos parámetros, ya que si se es demasiado conservador se puede dar el caso de que nunca llegue a haber instancias online debido a que no se les deja una ventana de tiempo suficiente para iniciarse y el balanceador las elimina y si se es demasiado generoso en la ventana de tiempo puede pasar que, como los recursos no aumentan, el sistema siga iniciando instancias, con lo que en vez de iniciar una instancia, nos podemos encontrar de golpe con tres o cuatro.

Para mostrar el funcionamiento, haremos uso del panel de AWS y nos conectaremos por SSH a cada instancia en servicio para mostrar el uso de CPU. Con la actual configuración e nuestro sistema, no es posible acceder a las instancias más de que a través del puerto 80 y a través del balanceador (recordemos que el puerto 443 está redirigido al 80 de las instancias. Por lo tanto, deberemos crear una regla en el grupo de seguridad de las instancias EC2 permitiendo la entrada por el puerto SSH (22) y deberemos asignar una IP (ElasticIP) a cada instancia para tener acceso directo. Esto sólo se hace para poder mostrar en detalle el funcionamiento del sistema de Autoescalado. No se debe dejar el puerto SSH abierto ni asignar direcciones IP a las máquinas en producción.

4.3.4.1.4.2 Estado inicial

Una vez configurado, el sistema se encarga de “levantar” el mínimo de instancias configuradas. A partir del momento en el que dichas instancias comiencen a operar, el sistema de auto-escalado empezará a monitorizar los recursos.

En las siguientes imágenes se muestra el sistema en el estado inicial:

| <input type="checkbox"/> | Name | Launch Configuration | Instances | Desired | Min | Max | Availability Zones |
|--------------------------|-----------------|-------------------------|-----------|---------|-----|-----|------------------------|
| <input type="checkbox"/> | Magento-Auto... | Magento-Launch-Confi... | 2 | 2 | 2 | 8 | eu-west-1a, eu-west-1b |

Ilustración 51: Situación inicial Auto-escalado

En la imagen anterior, vemos el panel de control del sistema de auto-escalado, que nos indica el número de instancias activas, las deseadas (en caso de que se dispare alguna alarma pero todavía no se hayan podido levantar la instancia), el número mínimo de instancias y el número máximo.

| Filter: State is OK | | | Search Alarms | |
|--|--|------------------------------------|---------------|--|
| State | Name | Threshold | | |
| <input type="checkbox"/> OK | awsec2-Magento-Autoscalling-Group-CPU-Utilization | CPUUtilization >= 60 for 5 minutes | | |
| <input checked="" type="checkbox"/> OK | awsec2-Magento-Autoscalling-Group-High-CPU-Utilization | CPUUtilization <= 30 for 5 minutes | | |

Ilustración 52: Situación inicial Auto-escalado. Alarmas OK.

En esta otra imagen vemos el sistema en estado inicial, con dos instancias web ejecutándose y ninguna alarma encendida.

| Instance ID | Name | Availability Zone | Status |
|-------------|------|-------------------|-------------|
| i-a7bfa0a | | eu-west-1b | InService ⓘ |
| i-b43f7b15 | | eu-west-1a | InService ⓘ |

Ilustración 53: Situación inicial Autoescalado. Elastic Load Balancer.

Por último, vemos cómo en el balanceador aparecen las instancias y están pasando el “Healthy Check”, ya que su estado es “En Servicio”.

Partiendo de este punto, iniciamos el mismo test que en pruebas anteriores, solo que, dadas las ventanas de tiempo que se configuran para el correcto funcionamiento del sistema de auto-escalado, el test tiene que durar más que los 240 segundos configurados anteriormente. Por ello, configuramos el test de forma que se ejecute de forma consecutiva diez veces, aumentando el tiempo de ejecución hasta los 40 minutos.

4.3.4.1.4.3 Iniciando test de estrés

Una vez iniciado el test, podemos observar cómo afecta a las instancias y a las métricas en las que se basan las alarmas:

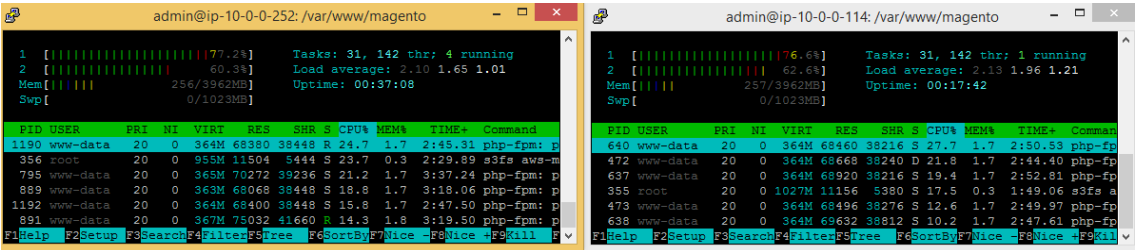


Ilustración 54: Auto-escalado.Carga servidor Htop.

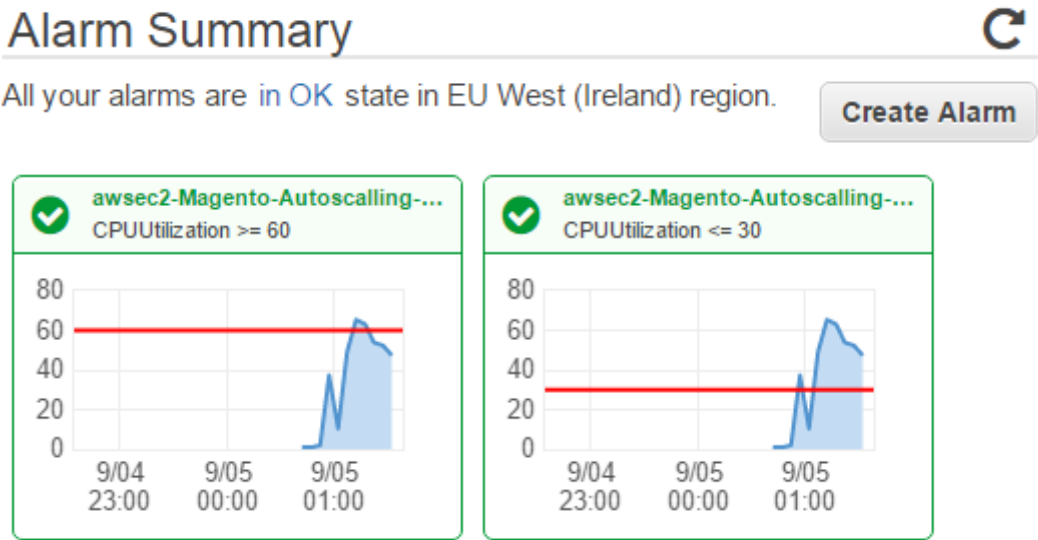


Ilustración 55: Auto-escalado. Alarmas OK.

Podemos ver cómo en los dos casos se está haciendo un uso alto de los recursos y cómo, aunque no se ha superado todavía durante el tiempo necesario como para activar la alarma, se han llegado a consumir más del 60% de los recursos CPU de las instancias.

4.3.4.1.4.4 Incrementando recursos

A medida que el test de stress incrementa el número de usuarios concurrentes, el sistema consume más recursos y, por tanto, necesita agregar más instancias.

4.3.4.1.4.4.1 Primer incremento

Al superarse durante más de 5 minutos consecutivos el límite de uso de recursos, se activa la alarma que inicia el incremento de los mismos:

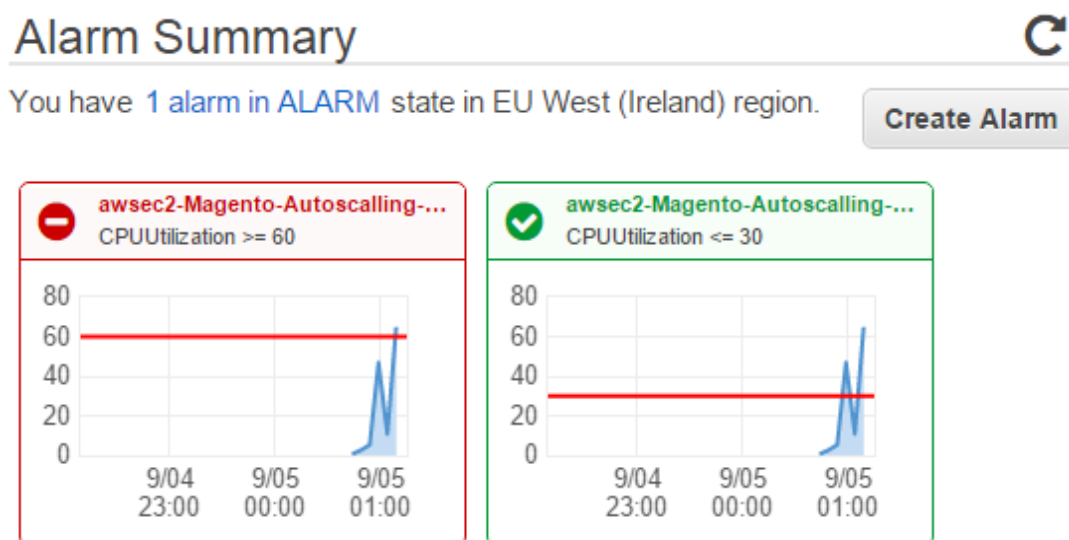


Ilustración 56: Auto-escalado. Alarma incremento de recursos disparada.

En el caso de que no hayamos llegado al límite de instancias activas permitidas de forma simultánea, de disparará la regla que incrementa el número de instancias. En función de la configuración, se puede incrementar en una o más instancias. En nuestro caso, una nueva instancia es agregada al balanceador y se queda a la espera de pasar el “Healthy Check”:

| Instance ID | Name | Availability Zone | Status |
|-------------|------|-------------------|----------------|
| i-a7bfa0a | | eu-west-1b | InService ⓘ |
| i-b43f7b15 | | eu-west-1a | InService ⓘ |
| i-df45017e | | eu-west-1a | OutOfService ⓘ |

Ilustración 57: Auto-escalado. ELB con instancia pendiente de healthy check.

Debemos recordar que tenemos configurada una ventana de 450 segundos desde que la instancia es iniciada hasta que el balanceador comienza a ejecutar los test. De esta forma nos aseguramos de que la instancia ha tenido tiempo de iniciarse y estar operativa y, si los test fallan, es que algo no ha ido bien y se debe eliminar esta instancia y crear una nueva.

Mientras la instancia no sea catalogada como “Saludable”, no recibirá tráfico web:

```

admin@ip-10-0-0-252: /var/www/magento
1 [||||||||||||||||| 62.6%] Tasks: 31, 121 thr: 1 running
2 [||||||||||||||||| 52.7%] Load average: 2.25 1.86 1.16
Mem[||||| 254/3962MB] Uptime: 00:39:00
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1192 www-data 20 0 363M 67568 38448 D 36.3 1.7 3:11.53 php-fpm: p
889 www-data 20 0 363M 68036 38448 S 23.4 1.7 3:42.90 php-fpm: p
1190 www-data 20 0 364M 68400 38448 S 22.9 1.7 3:08.89 php-fpm: p
356 root 20 0 953M 11496 5444 S 17.7 0.3 2:49.11 s3fs aws-m
795 www-data 20 0 365M 70024 39236 S 12.9 1.7 4:01.17 php-fpm: p
891 www-data 20 0 367M 75032 41660 S 12.4 1.8 3:44.21 php-fpm: p

admin@ip-10-0-0-114: /var/www/magento
1 [||||||||||||||||| 71.7%] Tasks: 31, 142 thr: 4 running
2 [||||||||||||||||| 56.9%] Load average: 1.81 1.98 1.31
Mem[||||| 256/3962MB] Uptime: 00:19:34
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
638 www-data 20 0 364M 69116 38812 R 32.6 1.7 3:11.93 php-fp
472 www-data 20 0 363M 69180 38240 S 29.7 1.7 3:08.46 php-fp
640 www-data 20 0 364M 68644 38216 S 23.4 1.7 3:13.81 php-fp
637 www-data 20 0 365M 68916 38216 D 20.4 1.7 3:16.81 php-fp
355 root 20 0 1027M 11160 5380 S 16.5 0.3 2:06.79 s3fs a
473 www-data 20 0 365M 69264 38276 S 11.7 1.7 3:14.56 php-fp

admin@ip-10-0-0-107: ~
1 [ 0.0%] Tasks: 25, 55 thr: 1 running
2 [ 0.0%] Load average: 0.08 0.07 0.05
Mem[| 62/3962MB] Uptime: 00:06:29
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
598 admin 20 0 24304 3492 2948 R 0.5 0.1 0:00.03 htop
1 root 20 0 28568 4632 3096 S 0.0 0.1 0:01.72 /sbin/init
149 root 20 0 32960 3820 3540 S 0.0 0.1 0:00.19 /lib/systemd/s
156 root 20 0 40788 3280 2748 S 0.0 0.1 0:00.21 /lib/systemd/s

```

Ilustración 58: Auto-escalado. Instancia a la espera de ser activada por ELB.

Una vez que el ELB activa la nueva instancia, ésta empieza a recibir tráfico y, por tanto, comienza a descargar al resto de instancias de trabajo:

```

admin@ip-10-0-0-252: /var/www/magento
1 [||||||||||||||||| 81.0%] Tasks: 32, 144 thr: 5 running
2 [||||||||||||||||| 64.4%] Load average: 2.32 2.05 1.34
Mem[||||| 263/3962MB] Uptime: 00:41:46
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
1192 www-data 20 0 363M 67568 38448 S 23.8 1.7 3:48.77 php-fpm: p
891 www-data 20 0 367M 75032 41660 S 23.8 1.8 4:19.39 php-fpm: p
889 www-data 20 0 366M 70168 38448 S 22.3 1.7 4:18.26 php-fpm: p
795 www-data 20 0 365M 70272 39236 S 19.8 1.7 4:36.16 php-fpm: p
356 root 20 0 955M 11552 5444 S 17.3 0.3 3:15.34 s3fs aws-m
1190 www-data 20 0 363M 67600 38448 S 12.9 1.7 3:44.49 php-fpm: p

admin@ip-10-0-0-114: /var/www/magento
1 [||||||||||||||||| 51.9%] Tasks: 30, 116 thr: 2 running
2 [||||||||||||||||| 38.5%] Load average: 2.18 2.04 1.44
Mem[||||| 251/3962MB] Uptime: 00:22:20
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
473 www-data 20 0 365M 68780 38276 S 37.4 1.7 3:53.03 php-fp
637 www-data 20 0 364M 68664 38216 S 23.7 1.7 3:56.68 php-fp
638 www-data 20 0 367M 71480 38812 S 15.6 1.8 3:49.90 php-fp
640 www-data 20 0 365M 69740 38216 S 13.3 1.7 3:50.68 php-fp
472 www-data 20 0 364M 68924 38240 S 11.8 1.7 3:43.90 php-fp
355 root 20 0 1027M 11092 5380 S 10.4 0.3 2:32.46 s3fs a

admin@ip-10-0-0-107: /var/www/magento
1 [||||||||||||||||| 53.9%] Tasks: 30, 116 thr: 3 running
2 [||||||||||||||||| 42.2%] Load average: 1.35 0.37 0.16
Mem[||||| 237/3962MB] Uptime: 00:09:16
Swp[| 0/1023MB]

PID USER PRI NI VIRT RES SHR S CPU% MEM% TIME+ Command
563 www-data 20 0 364M 67740 37648 R 28.8 1.7 0:04.54 php-fpm: pool
557 www-data 20 0 363M 66594 37728 S 27.3 1.6 0:05.11 php-fpm: pool
558 www-data 20 0 376M 79648 37480 R 22.9 2.0 0:04.60 php-fpm: pool
634 www-data 20 0 363M 66492 37348 S 22.0 1.6 0:04.19 php-fpm: pool

```

Ilustración 59: Auto-escalado. Todas las instancias recibiendo tráfico.

En el panel de Auto-escalado, podemos ver cómo se ha reflejado el cambio en el número de instancias activas:

| <input type="checkbox"/> | Name | Launch Configuration | Instances | Desired | Min | Max | Availability Zones |
|--------------------------|-----------------|-------------------------|-----------|---------|-----|-----|------------------------|
| <input type="checkbox"/> | Magento-Auto... | Magento-Launch-Confi... | 3 | 3 | 2 | 8 | eu-west-1a, eu-west-1b |

Ilustración 60: Auto-escalado. Panel auto-escalado con tres instancias.

4.3.4.1.4.4.2 Segundo incremento

A pesar de la nueva instancia se sigue superando el límite de uso de la CPU, por lo que se incrementa de nuevo el número de instancias, pasando a cuatro. En este momento ya podemos ver cómo la carga de la CPU se mantiene por debajo del 60%

| Instance ID | CPU % | Mem % | Tasks | Running |
|---------------|-------|------------|-------------|---------|
| ip-10-0-0-252 | 49.0% | 262/3962MB | 31, 142 thr | 3 |
| ip-10-0-0-114 | 54.5% | 250/3962MB | 31, 142 thr | 5 |
| ip-10-0-0-107 | 57.0% | 256/3962MB | 31, 142 thr | 2 |
| ip-10-0-0-146 | 51.0% | 233/3962MB | 30, 116 thr | 1 |

Ilustración 61: Auto-escalado. Carga CPU cuatro instancias.

En el panel del ELB podemos ver las cuatro instancias, todas “Saludables”:

| Instance ID | Name | Availability Zone | Status |
|-------------|------|-------------------|-------------|
| i-a7befa0a | | eu-west-1b | InService ⓘ |
| i-b43f7b15 | | eu-west-1a | InService ⓘ |
| i-73c387de | | eu-west-1b | InService ⓘ |
| i-df45017e | | eu-west-1a | InService ⓘ |

Ilustración 62: Auto-escalado. ELB con 4 instancias.

Si nos fijamos en el panel de instancias EC2, podemos ver las cuatro instancias web que componen actualmente nuestro sistema, dos en cada zona de disponibilidad:

| | | | | | |
|--------------------------|------------|-----------|------------|-----------|-----------------|
| <input type="checkbox"/> | i-73c387de | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks... |
| <input type="checkbox"/> | i-a7befa0a | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks... |
| <input type="checkbox"/> | i-b43f7b15 | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks... |
| <input type="checkbox"/> | i-df45017e | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks... |

Ilustración 63: Auto-escalado. Panel EC2 con 4 instancias.

4.3.4.1.4.5 Decrementando recursos

En el momento en el que el uso de los recursos se sitúe por debajo del 30% la alarma correspondiente se activará. En caso de que el sistema no tenga activas ya el número mínimo de instancias, una de ellas será eliminada.

Para poder realizar esta prueba hemos iniciado el mismo test que para el incremento de recursos, pero hemos reducido el número de usuarios simultáneos de 40 a 10.

| Instance ID | IP | CPU% | Mem | Tasks | Uptime |
|-------------|------------|-------|------------|------------------------|----------|
| i-73c387de | 10-0-0-252 | 24.4% | 181/3962MB | 27, 91 thr: 1 running | 00:52:24 |
| i-a7befa0a | 10-0-0-114 | 23.3% | 178/3962MB | 27, 90 thr: 2 running | 00:32:56 |
| i-b43f7b15 | 10-0-0-107 | 13.0% | 178/3962MB | 27, 91 thr: 4 running | 00:19:51 |
| i-df45017e | 10-0-0-146 | 25.1% | 205/3962MB | 29, 116 thr: 1 running | 00:11:51 |

Ilustración 64: Auto-escalado. Baja carga CPU cuatro instancias.

En la imagen anterior podemos ver cómo el uso de CPU de las instancias

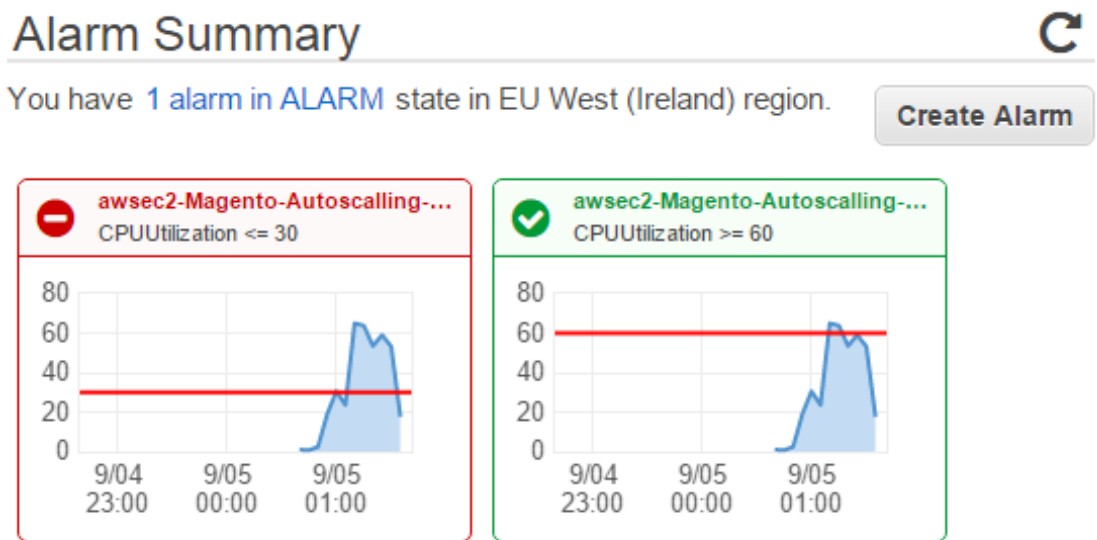


Ilustración 65: Auto-escalado. Alarma decremento de recursos disparada.

La alarma disparará la regla correspondiente, que se encargará de reducir en una unidad el número de instancias disponibles. Podemos ver en la siguiente imagen cómo se ejecuta la orden en el panel de instancias EC2:

| | | | | | |
|-------------------------------------|------------|-----------|------------|------------------|-----------------|
| <input type="checkbox"/> | i-73c387de | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks... |
| <input type="checkbox"/> | i-a7bfa0a | t2.medium | eu-west-1b | ● running | ✓ 2/2 checks... |
| <input checked="" type="checkbox"/> | i-b43f7b15 | t2.medium | eu-west-1a | ● shutting-do... | |
| <input type="checkbox"/> | i-df45017e | t2.medium | eu-west-1a | ● running | ✓ 2/2 checks... |

Ilustración 66: Auto-escalado. Panel EC2 con eliminando instancia-1.

En caso de que no sea suficiente y el uso de CPU se sitúe por debajo del 30%, la alarma de decremento de recursos seguirá activa, pero será ignorada hasta que transcurran los 600 segundos de ventana impuestos en la configuración para dar tiempo al sistema a que se estabilice.

En caso de que transcurrido ese tiempo la alarma de decremento de recursos siga activada, se volverá a ejecutar:

| | | | | | |
|--------------------------|------------|-----------|------------|----------------|---------------|
| | i-73c387de | t2.medium | eu-west-1b | shutting-do... | |
| <input type="checkbox"/> | i-a7bfa0a | t2.medium | eu-west-1b | running | 2/2 checks... |
| <input type="checkbox"/> | i-b43f7b15 | t2.medium | eu-west-1a | terminated | |
| <input type="checkbox"/> | i-df45017e | t2.medium | eu-west-1a | running | 2/2 checks... |

Ilustración 67: Auto-escalado. Panel EC2 con eliminando instancia-2.

Quedándose finalmente como al principio, con únicamente dos instancias ejecutándose al ser el mínimo número permitido.

| | | | | | |
|--------------------------|------------|-----------|------------|------------|---------------|
| <input type="checkbox"/> | i-73c387de | t2.medium | eu-west-1b | terminated | |
| <input type="checkbox"/> | i-a7bfa0a | t2.medium | eu-west-1b | running | 2/2 checks... |
| <input type="checkbox"/> | i-b43f7b15 | t2.medium | eu-west-1a | terminated | |
| <input type="checkbox"/> | i-df45017e | t2.medium | eu-west-1a | running | 2/2 checks... |

Ilustración 68: Auto-escalado. Panel EC2 con eliminando instancia-3.

Es probable que se dé el caso de que la alarma que detecta un uso inferior al 30% de la CPU siga activa debido a que se hace un uso limitado del sistema. Al haberse llegado al número mínimo de instancias, no se ejecutará la regla de decremento de recursos.

4.3.4.1.5 Prueba acceso desde diferentes países

En este apartado testaremos una de las funcionalidades más importantes de nuestra tienda, que es el hecho de muestre la tienda adecuada en función de la localización geográfica del cliente.

Recordemos que esto es esencial para poder prestar un servicio de envío global y con bajo coste. Por ello, se establecieron tres zonas diferentes (Europa, América y Asia) de forma que cada una de ellas tiene un almacén de referencia desde el que se envían los pedidos a los países de esa zona.

Para poder realizar el test, debemos hacer uso de servidores proxy [161] situados en países ubicados en diferentes zonas. Los países elegidos son España, Estados Unidos y Japón. Usaremos los servidores proxy de pago de un conocido servicio de internet que configuraremos en nuestro navegador para que las peticiones de acceso a la web tengan origen en el país que nos interese.

Para cada prueba configuraremos el servidor proxy del país que necesitemos y recargaremos la página. Para facilitar la configuración, haremos uso de un plugin para el navegador Chrome. El orden de pruebas será el siguiente:

- La primera prueba se hará desde España, de forma que la configuración del proxy debe estar desactivada para que las peticiones sean hechas desde nuestra ubicación. Se deberá cargar la tienda europea, es decir “Europe Island”.
- La segunda prueba se hará desde Estados Unidos, por lo que debemos configurar el plugin del navegador para que se conecte al proxy de Estados Unidos. Se deberá cargar la tienda americana, es decir “America Island”.
- La tercera prueba se hará desde Japón, por lo que debemos configurar el plugin del navegador para que se conecte al proxy de Japón. Se deberá cargar la tienda asiática, es decir “Asia Island”.

4.3.4.1.5.1 Website Europa

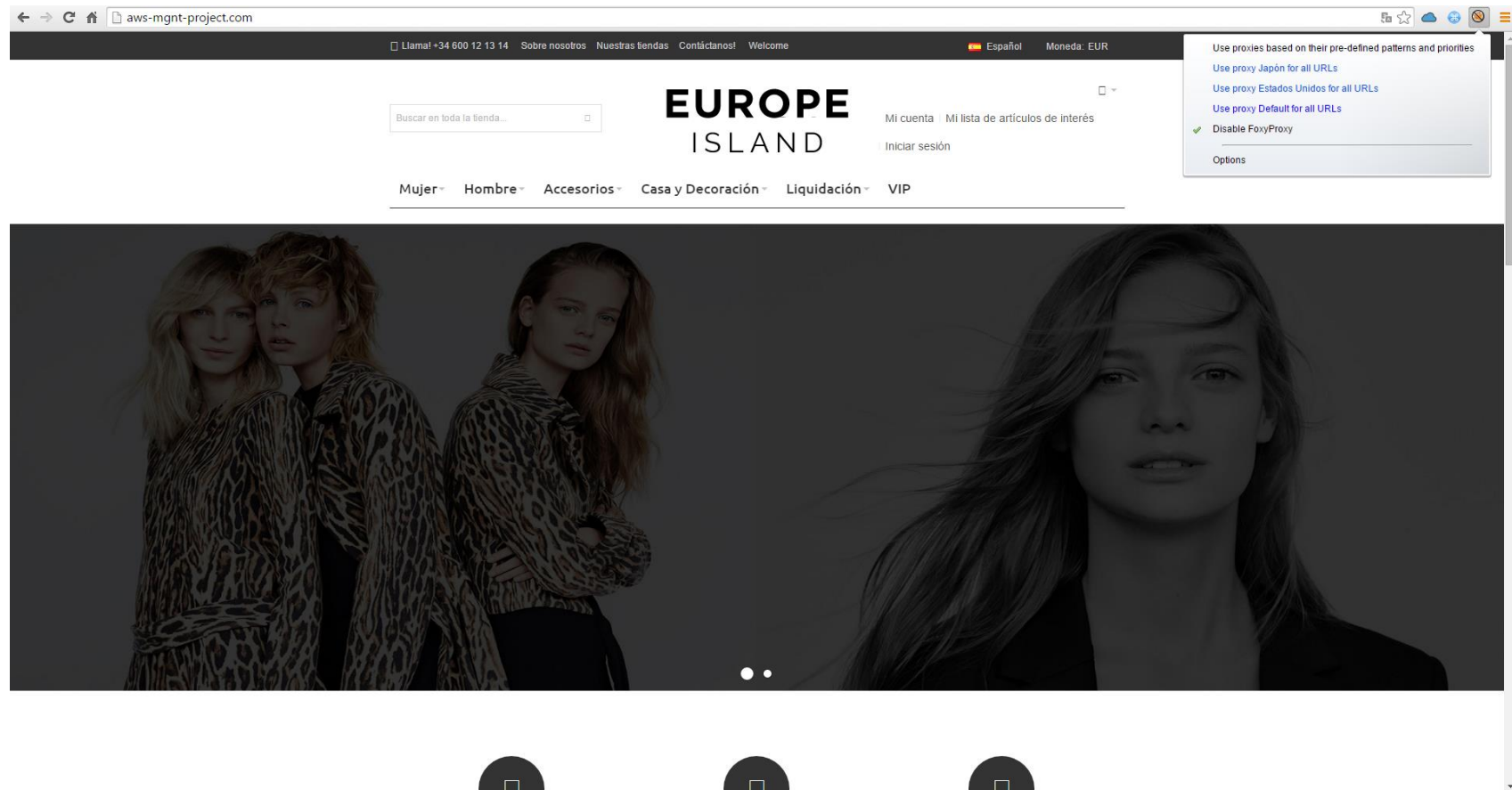


Ilustración 69: Home Tienda Europa Island

4.3.4.1.5.2 Website America

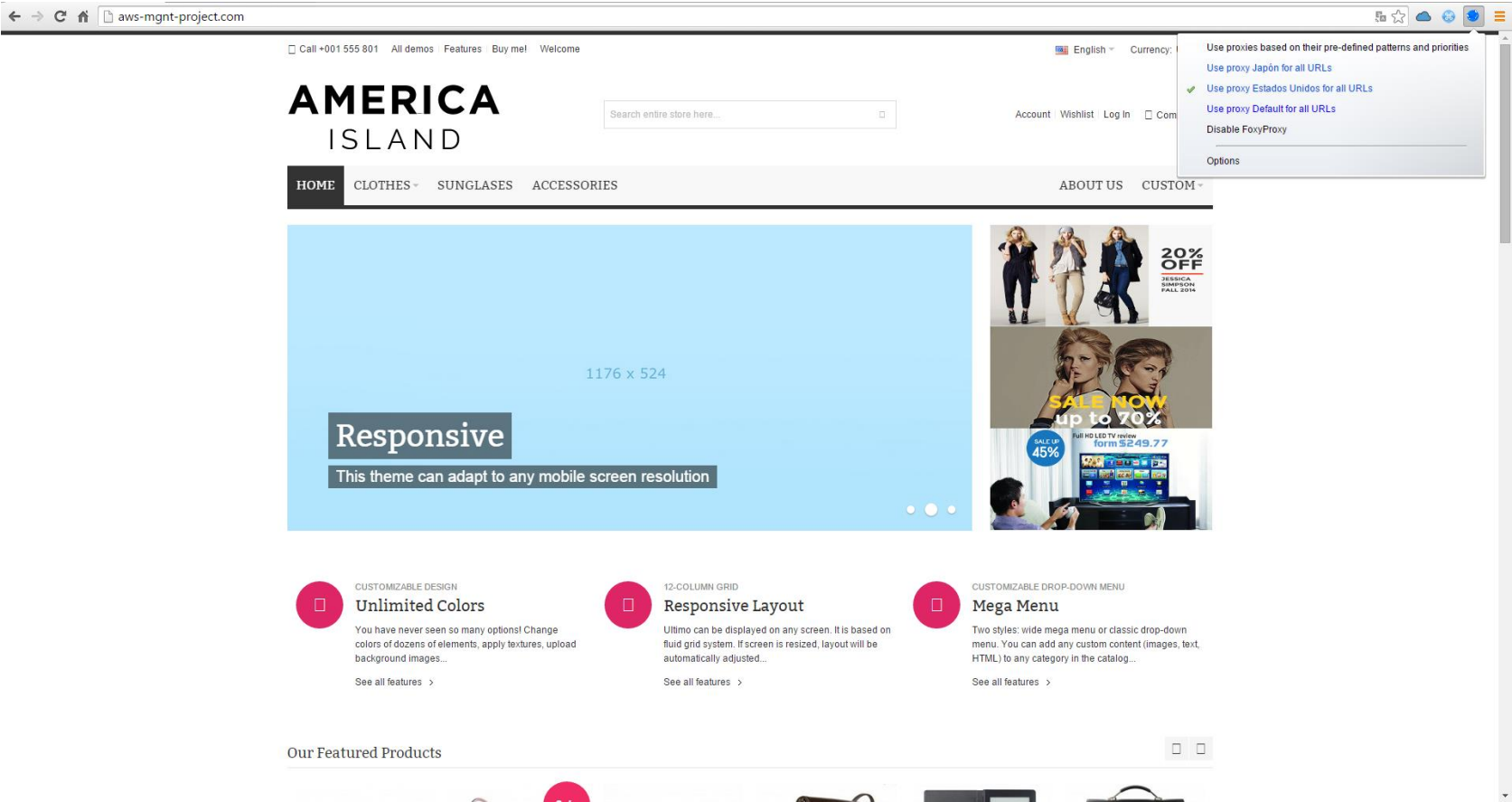


Ilustración 70: Home Tienda America Island

4.3.4.1.5.3 Website Asia

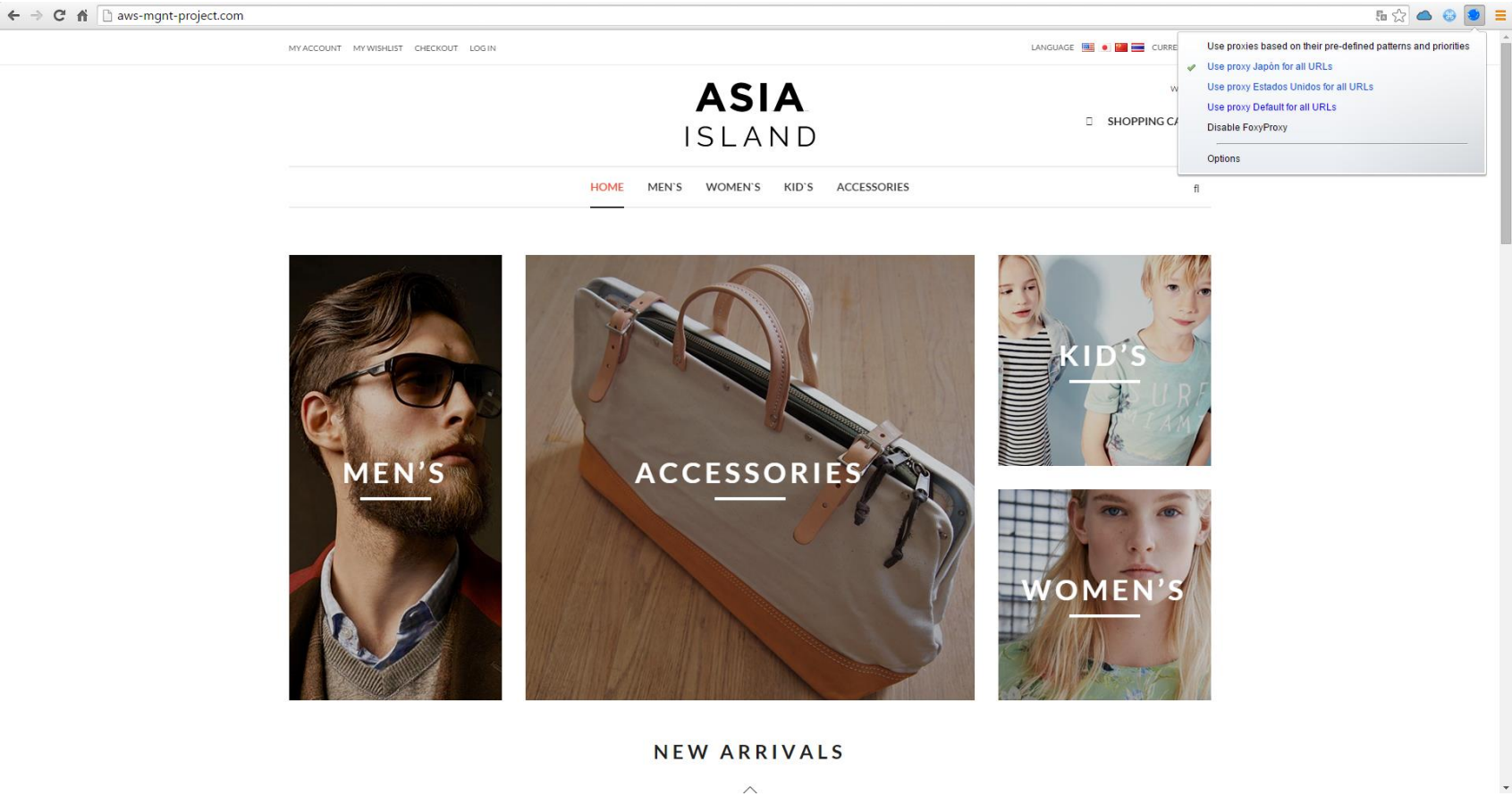


Ilustración 71: Home Tienda Asia Island

4.3.4.1.6 Conclusiones

A través de las diferentes pruebas realizadas hemos podido testear los sistemas de caché y de mejora del rendimiento de Magento así como su impacto en los diferentes recursos hardware (web, base de datos, etc).

También hemos podido verificar que el sistema S3FS como servicio para compartir archivos entre distintas instancias web es extremadamente lento a la hora de servir contenidos web (css, js e imágenes), aumenta el consumo de recursos en las instancias web y utiliza un alto número de peticiones GET, por las cuales, pasado un límite AWS comienza a cobrar. Por ello es necesario el uso de un CDN de forma que mantenemos su funcionalidad como sistema de compartición de archivos para las instancias web pero será el CDN el que se encargue de su distribución web.

En lo referente a las instancias web hemos visto que, como norma general, incluso en los momentos en los que el uso de CPU es muy alto cada instancia dispone de una gran cantidad de memoria RAM libre. Claramente estamos desperdiciando recursos por lo que se debe actuar en la configuración de las instancias.

La cantidad de RAM viene determinada por el tipo de instancia que elijamos y no parece razonable hacer uso de una instancia de menor rendimiento, por lo que debemos buscar la forma de utilizar la memoria RAM excedente. Opciones:

- Ajustar la configuración de PHP-FPM de forma que haga mayor uso de la RAM.
- Activar diferentes optimizadores / cachés de código PHP.
- Activar el uso de memoria caché distribuida (Memcached).

Aunque hasta el momento no ha ocasionado problemas, hemos de recordar que el acceso a disco y el uso de la red están limitados. El uso de disco, haciendo uso cada vez más de sistemas caché en red o memoria, no nos debe preocupar especialmente, pero sí el uso de red.

Es de vital importancia ajustar correctamente la configuración de auto-escalado para asegurar un rendimiento óptimo y coste razonable. Si de forma habitual el sistema debe “levantar” instancias extra, debemos plantearnos la posibilidad de incrementar los recursos de las instancias “base”.

4.3.4.2 Validación de los requisitos de usuario

En este apartado verificaremos el cumplimiento de los requisitos establecidos en el apartado “3.1.1 Requisitos de usuario”. La validación se hará mediante la prueba o ejecución de dicho requisito de forma que se pueda evaluar si se ha cumplido con su correcta implantación.

Para la definición de cada uno de las pruebas haremos uso de una “tabla plantilla” como la siguiente:

| Identificador de validación de requisito | |
|--|---|
| Descripción | |
| Prueba | |
| Resultado | <input type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 111: Plantilla validación requisitos

Descripción de los campos:

- Identificador de validación de requisito: Campo único que identifica la validación del requisito de usuario. La nomenclatura que se usará será V-RU-XXX
- Descripción: Descripción del requisito
- Prueba: Acciones realizadas para validar el requisito
- Resultado: Puede tomar dos valores
 - Correcto: El requisito de usuario se cumple en el sistema creado.
 - Incorrecto: El requisito de usuario no se cumple en el sistema creado.

| V-RU-001 | |
|--------------------|---|
| Descripción | El usuario podrá elegir el idioma en el que visualizar la tienda entre varias opciones disponibles |
| Prueba | <ol style="list-style-type: none">1. Se accede a la tienda.2. Se despliega el selector de idiomas y se elige una de las opciones.3. Se comprueba que se recarga correctamente la página mostrando la tienda en el idioma elegido. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 112: Validación de requisito de usuario 001

| V-RU-002 | |
|--------------------|--|
| Descripción | El usuario podrá elegir la moneda en la que visualizar los precios de los productos entre varias opciones. |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se despliega el selector de monedas y se elige una de las opciones. 3. Se comprueba que se recarga correctamente la página mostrando los precios de la tienda en la moneda elegida. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 113: Validación de requisito de usuario 002

| V-RU-003 | |
|--------------------|--|
| Descripción | El usuario podrá registrarse en el sistema. |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se selecciona "Registrarse". 3. Se completa el formulario de registro y se envía. 4. Se comprueba que hemos iniciado sesión en el sistema accediendo a la "Zona de cliente", por lo que el registro ha sido satisfactorio. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 114: Validación de requisito de usuario 003

| V-RU-004 | |
|--------------------|--|
| Descripción | El usuario podrá iniciar sesión en el sistema |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se selecciona "Iniciar sesión". 3. Se introduce el nombre de usuario y contraseña y se envía. 4. Se comprueba que hemos iniciado sesión en el sistema accediendo a la "Zona de cliente". |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 115: Validación de requisito de usuario 004

| V-RU-005 | |
|--------------------|---|
| Descripción | El usuario registrado podrá visualizar su historial de pedidos. |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se selecciona “Iniciar sesión”. 3. Se introduce el nombre de usuario y contraseña y se envía. 4. Se accede a la zona de cliente y se selecciona “Historial de pedidos” 5. Se comprueba que se muestra correctamente el historial de pedidos. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 116: Validación de requisito de usuario 005

| V-RU-006 | |
|--------------------|--|
| Descripción | El usuario podrá recuperar su carrito de la compra durante la duración de su cookie de sesión |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se añade un producto al carro de la compra. 3. Se selecciona otra web para navegar, distinta a la de la tienda, y se navega por la misma durante unos minutos. 4. Se vuelve a la tienda y se accede al carro de la compra. 5. Se comprueba que permanece en el carro de la compra el producto añadido con anteriormente |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 117: Validación de requisito de usuario 006

| V-RU-007 | |
|--------------------|--|
| Descripción | El usuario podrá comprar productos |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se añade un producto al carro de la compra. 3. Se accede al checkout y se rellenan los datos de pago si ya estamos registrados y los de cliente si no estamos registrados. 4. Finalizamos la compra y efectuamos el pago según la opción elegida. 5. Recibimos un email confirmándonos la recepción del pedido y otro confirmándonos el pago. |

| | | |
|------------------|--|-------------------------------------|
| Resultado | <input checked="" type="checkbox"/> Correcto | <input type="checkbox"/> Incorrecto |
|------------------|--|-------------------------------------|

Tabla 118: Validación de requisito de usuario 007

| V-RU-008 | |
|--------------------|---|
| Descripción | El usuario podrá acceder a la tienda desde su Smartphone con plena funcionalidad |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda desde un Smartphone. 2. Se navega por la tienda accediendo a las páginas de categoría y producto, comprobando que se visualiza todo correctamente. 3. Se añade un producto al carro de la compra. 4. Se accede al checkout y se rellenan los datos de pago si ya estamos registrados y los de cliente si no estamos registrados. 5. Finalizamos la compra y efectuamos el pago según la opción elegida. 6. Se recibe un email confirmándonos la recepción del pedido y otro confirmándonos el pago. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 119: Validación de requisito de usuario 008

| V-RU-009 | |
|--------------------|--|
| Descripción | El usuario podrá acceder a la tienda desde su tablet con plena funcionalidad |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda desde una tablet. 2. Se navega por la tienda accediendo a las páginas de categoría y producto, comprobando que se visualiza todo correctamente. 3. Se añade un producto al carro de la compra. 4. Se accede al checkout y se rellenan los datos de pago si ya estamos registrados y los de cliente si no estamos registrados. 5. Finalizamos la compra y efectuamos el pago según la opción elegida. 6. Se recibe un email confirmándonos la recepción del pedido y otro confirmándonos el pago. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 120: Validación de requisito de usuario 009

| V-RU-010 | |
|--------------------|---|
| Descripción | El usuario podrá comparar productos |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se accede a la página de producto o a la de categoría y se elige la opción “Añadir para comparar” en un producto. 3. Se repite la acción anterior con otro producto y somos redirigidos a la página de comparación de los productos en la que podemos visualizar desglosados los dos productos elegidos. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 121: Validación de requisito de usuario 010

| V-RU-011 | |
|--------------------|--|
| Descripción | El usuario podrá añadir productos a su lista de deseos |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se accede a la página de producto o a la de categoría y se elige la opción “Añadir a la lista de deseos” en un producto. 3. Se inicia sesión. 4. Se accede, dentro de la zona privada de cliente, al apartado “Mi lista de deseos”. 5. Se comprueba que aparece el producto añadido a la lista. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 122: Validación de requisito de usuario 011

| V-RU-012 | |
|--------------------|--|
| Descripción | El usuario podrá realizar búsquedas de productos |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. En el buscador se escribe “pantalón”. 3. Como resultado nos devuelve varios productos cuyo nombre que coinciden totalmente o en parte con la palabra buscada. |

| | |
|------------------|--|
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |
|------------------|--|

Tabla 123: Validación de requisito de usuario 012

| V-RU-013 | |
|--------------------|---|
| Descripción | El usuario tendrá acceso a la política de envíos |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. En el footer, se hace “click” en el enlace “Política de envíos”. 3. Se comprueba que se recarga la página y accedemos a la política de envíos. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 124: Validación de requisito de usuario 013

| V-RU-014 | |
|--------------------|---|
| Descripción | El usuario tendrá acceso a la política de privacidad |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. En el footer, se hace “click” en el enlace “Política de privacidad”. 3. Se comprueba que se recarga la página y accedemos a la política de privacidad. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 125: Validación de requisito de usuario 014

| V-RU-015 | |
|--------------------|--|
| Descripción | El usuario tendrá acceso a la política de devoluciones |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. En el footer, se hace “click” en el enlace “Política de devoluciones”. <p>Se comprueba que se recarga la página y accedemos a la política de devoluciones.</p> |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 126: Validación de requisito de usuario 015

| V-RU-016 | |
|--------------------|--|
| Descripción | El usuario tendrá acceso a la política de cookies |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. En el footer, se hace “click” en el enlace “Política de cookies”. <p>Se comprueba que se recarga la página y accedemos a la política de cookies.</p> |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 127: Validación de requisito de usuario 016

| V-RU-017 | |
|--------------------|--|
| Descripción | El usuario se podrá suscribir a la newsletter de la tienda |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se accede a la sección o banner de newsletter. 3. Se introduce un correo electrónico y se envía. 4. Se accede al backoffice de Magento. 5. En la sección de newsletter se comprueba que el email se ha añadido correctamente. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 128: Validación de requisito de usuario 017

| V-RU-018 | |
|--------------------|---|
| Descripción | El usuario podrá navegar por las categorías de la tienda |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se navega por todas las categorías de la tienda. 3. Se comprueba que todas son accesibles y se muestran correctamente. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 129: Validación de requisito de usuario 018

| V-RU-019 | |
|--------------------|---|
| Descripción | El usuario se podrá visualizar todos los productos de la tienda |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se accede a todos los productos de la tienda entrando |

| | |
|------------------|--|
| | <p>una a una en todas las categorías.</p> <p>3. Se comprueba que todos los productos son accesibles y se muestran correctamente.</p> |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 130: Validación de requisito de usuario 019

| V-RU-020 | |
|--------------------|---|
| Descripción | El usuario podrá comprar los productos desde la página de catálogo |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se accede a una categoría. 3. Se comprueba que, desde la página de la categoría, cada producto dispone de un “botón” que nos permite la compra del mismo. 4. Se selecciona dicho “botón”. 5. Se comprueba que el producto se ha añadido al carro de la compra. 6. Se repite el proceso con varios productos y en varias categorías. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 131: Validación de requisito de usuario 020

| V-RU-021 | |
|--------------------|--|
| Descripción | El usuario podrá ver en todo momento un resumen del carro de la compra |
| Prueba | <ol style="list-style-type: none"> 1. Se accede a la tienda. 2. Se añaden varios productos al carro de la compra. 3. Se navega por toda la web y, en cada página se comprueba que es visible el icono del carro de la compra que, si se hace “click” en él, se despliega y nos permite ver el resumen de la compra. |
| Resultado | <input checked="" type="checkbox"/> Correcto <input type="checkbox"/> Incorrecto |

Tabla 132: Validación de requisito de usuario 021

4.4 Problemas encontrados

A medida que se ha ido elaborando el proyecto, nos hemos ido encontrando con diferentes problemas que nos han obligado a buscar soluciones alternativas a las propuestas en un primer momento.

Un ejemplo de ello es la modificación en el diseño de la arquitectura del sistema auto-escalable. En un primer momento se estableció un servidor NFS como forma de que las instancias web compartieran los datos necesarios entre ellas, pero se vio que no era posible tener disponible un respaldo de dicho servidor, lo que convertía dicho servidor en un SPOF (Single Point of Failure) echando por tierra uno de los objetivos del sistema, la alta disponibilidad.

La solución a este problema pasó por la sustitución del NFS por un servidor Redis para la caché y las sesiones y por S3 para las imágenes y ficheros css y js. Para éste último, mediante el proyecto S3FS se hace uso del servicio S3 de AWS como si de un sistema de ficheros se tratase. Se esta forma, en cada instancia web se monta la capeta /media en “red”.

Todo cambio provoca reacciones nuevas e inesperadas. Descubrimos que este nuevo sistema acarrea una serie de problemas, tales como una alta latencia y coste, ya que la cantidad de peticiones “GET” y “PUT” a S3 que se generaban con la navegación de un solo usuario agotaban el “saldo” de este tipo de peticiones gratuitas.

La solución vino de la mano de la implantación de un sistema de distribución de contenidos (CDN) de manera que mantuviéramos S3 funcionando como un NFS, pero derivando la distribución de los mismos en el CDN. Al estar ya previsto en un principio, no provocó ninguna desviación en el proyecto.

Otro problema asociado al uso de S3FS fue el hecho de que, en determinadas situaciones, algunas instancias no montaban correctamente la unidad en red /media, lo que provocaba un problema de conexión. Esto en un entorno auto-escalable no es admisible, ya que bloquearía el sistema. Para solucionar este problema se integró la comprobación de la correcta conexión a S3FS en el “HealthyCheck” que realiza el ELB periódicamente para comprobar el correcto funcionamiento de cada instancia web. De esta forma, en caso de que al iniciarse la instancia la unidad /media no se hubiese montado correctamente, no superaría el test, el ELB la desecharía y “levantaría” una nueva.

Al ser una simple modificación del fichero de pruebas “healthycheck.php”, este problema no tampoco provocó desviación alguna en el proyecto.

5. Gestión del proyecto

5.1 Planificación inicial

La planificación inicial del proyecto la reflejamos en el siguiente diagrama GANTT

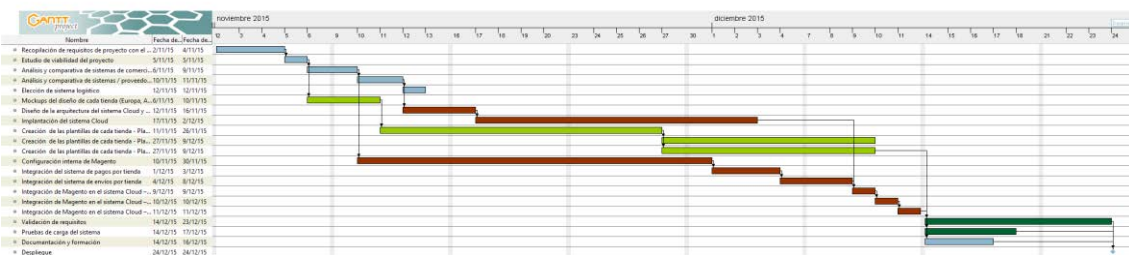


Ilustración 72: GANTT - Planificación inicial

Con su correspondiente ruta crítica:

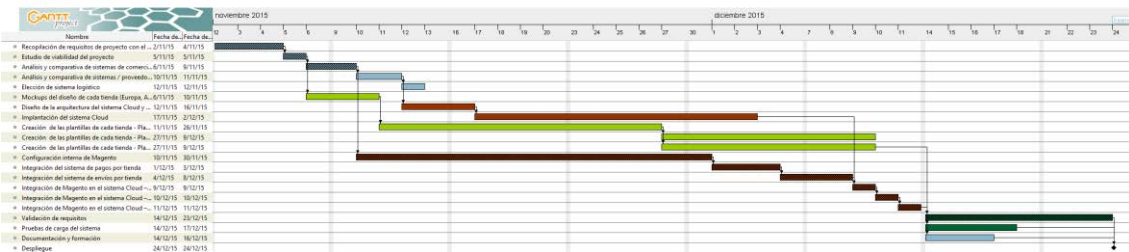


Ilustración 73: GANTT - Planificación inicial - camino crítico

5.2 Planificación final

Debido a los problemas encontrados mencionados anteriormente, el proyecto sufrió una desviación de dos días que reflejamos a continuación en el siguiente diagrama GANTT:

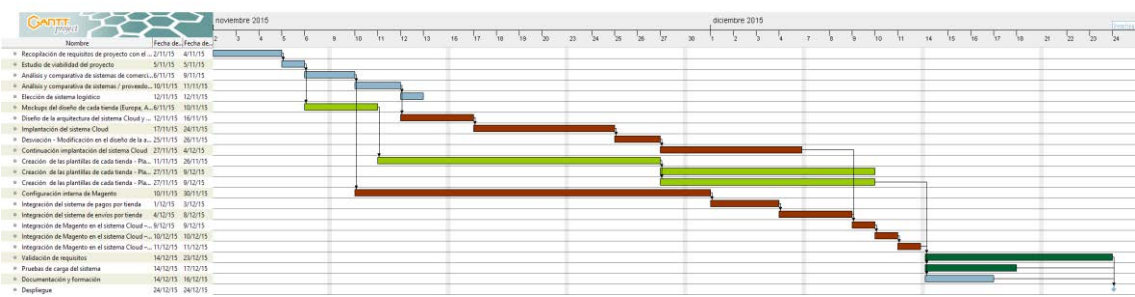


Ilustración 74: GANTT - Desviación planificación inicial

Dicha desviación no afectó a la fecha de entrega del proyecto al no encontrarse la tarea en la ruta crítica, como vemos a continuación:

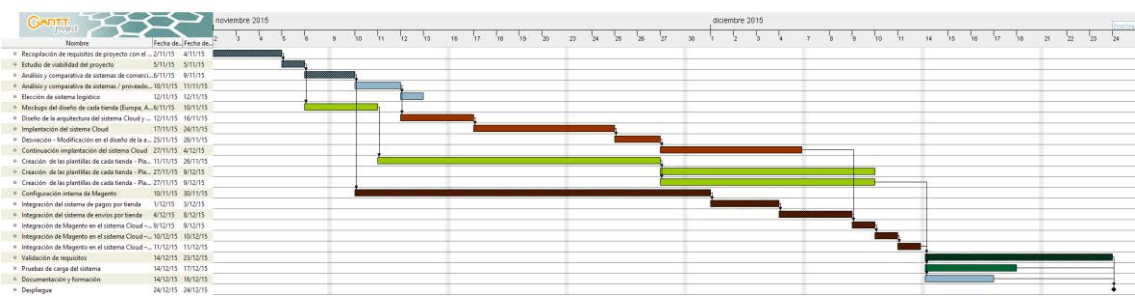


Ilustración 75: Desviación planificación inicial - camino crítico

5.3 Presupuesto

5.3.1 Presupuesto Desarrollo

A continuación presentamos el desglose de tareas, las horas que requieren y el equipo responsable:

| Presupuesto Desarrollo | | | | | | |
|------------------------|----|---|---------------------|-----|-----|-----|
| Fase | ID | Tarea | Responsable / Horas | | | |
| | | | G | D | P | T |
| Fase 1 | 1 | Recopilación de requisitos de proyecto con el cliente | 24 | | | |
| | 2 | Estudio de viabilidad del proyecto | 8 | | | |
| | 3 | Análisis y comparativa de sistemas de comercio electrónico | 16 | | | |
| | 4 | Análisis y comparativa de sistemas / proveedores Cloud | 16 | | | |
| | 5 | Elección de sistema logístico | 8 | | | |
| | 6 | Mockups del diseño de cada tienda (Europa, Asia, América) | | 24 | | |
| Fase 2 | 7 | Diseño de la arquitectura del sistema Cloud y elección de las tecnologías | | | 24 | |
| | 8 | Implantación del sistema Cloud | | | 96 | |
| | 9 | Creación de las plantillas de cada tienda | | 250 | | |
| | 10 | Configuración interna de Magento | | | 120 | |
| | 11 | Integración del sistema de pagos por tienda | | | 24 | |
| | 12 | Integración del sistema de envíos por tienda | | | 24 | |
| | 13 | Integración de Magento en el sistema Cloud – Asociación geolocalización y tienda | | | 8 | |
| | 14 | Integración de Magento en el sistema Cloud – Configuración Magento entorno autoescalable | | | 8 | |
| | 15 | Integración de Magento en el sistema Cloud – Configuración y gestión SSL en entorno autoescalable | | | 8 | |
| Fase 3 | 16 | Validación de requisitos | | | | 64 |
| | 17 | Pruebas de carga del sistema | | | | 32 |
| | 18 | Despliegue | | | 8 | |
| | 19 | Documentación y formación | 24 | | | |
| | 20 | Mantenimiento (6 meses) | | | | 48 |
| TOTAL | | | 96 | 274 | 320 | 144 |

Tabla 133: Presupuesto desarrollo

Leyenda:

- G -> Gestor de proyecto
- D -> Diseñador
- P - Programador
- T -> Equipo de testing

A continuación mostramos el resumen de horas y el coste de cada equipo:

| Coste Recursos Humanos | | | | |
|----------------------------|--------------------|-----------|-------------|----------------|
| | Gestor de proyecto | Diseñador | Programador | Equipo testing |
| Horas dedicadas | 96 | 274 | 320 | 144 |
| Precio recurso | 35€/h | 25€/h | 28€/h | 20€/h |
| Coste total recurso | 3360€ | 6850€ | 8960€ | 2880€ |

Tabla 134: Coste Recursos Humanos

5.3.2 Presupuesto tecnología AWS

En este apartado realizamos una tabla resumen de los servicios de AWS necesarios para este proyecto y su coste en función de un cálculo estimado de uso. Dicho cálculo puede variar dependiendo de las configuraciones finales, del tráfico (recordemos que es un sistema auto-escalable) y demás variables:

| Presupuesto sistema AWS | | | |
|-------------------------|--|------------------|-------------------|
| Servicio AWS | Detalles | Más Info | Coste mensual (€) |
| EC2 | Dos instancias t2.medium con 8GB SSD | N/A | 104,66\$ |
| ELB | Un balanceador web con gestión SSL | 50GB tráfico/mes | Inc. en EC2 |
| S3 | Almacenamiento de Snapshots, AMI, carpeta /media y Backups | 15GB | 0.5\$ |
| AutoScalling | Sistema automático de autoescalado de recursos web | N/A | N/A |
| RDS | Instancia MySQL db.m3-medium | Multi-AZ | 146,06\$ |
| ElasticCache | Instancia Redis cache.m3.medium | Multi-AZ | 139,08\$ |

| | | | |
|-------------------|---|--------------------------------|----------|
| Route53 | Gestión dominioa aws-mgnt.project.com e internal.aws-mgnt.project.com | 2 zonas. 10 millones consultas | 3\$ |
| CloudWatch | Seguimiento instancias y gestión de alarmas | Capa gratuita | 0\$ |
| CodeDeploy | No se utiliza por el momento | N/A | N/A |
| VPC | VPC Mgnt-Proyect. Dos zonas de disponibilidad (A y B) | N/A | N/A |
| CloudFront | Sistema CDN distribuido por todo el mundo | 50GB/mes | 5,65\$ |
| SES | Servidor SMTP para el envío de emails | 50 emails / día 5GB/mes | 1\$ |
| Data Transfer IN | Transferencia de datos desde internet | 200GB/mes | 0\$ |
| Data Transfer OUT | Transferencia de datos hacia internet | 200GB/mes | 20\$ |
| TOTAL | Coste mensual Sistema Cloud en AWS | | 419.95\$ |

Tabla 135: Presupuesto sistema AWS

5.3.3 Presupuesto Final

En la siguiente tabla presentamos el coste final del proyecto:

| Presupuesto Total | | |
|-------------------------|------------|---------------------------|
| Servicio AWS | Coste | Coste mensual (€) Sin IVA |
| Presupuesto Desarrollo | Pago único | 22.050€ |
| Presupuesto sistema AWS | Mensual | 419,95\$ |

Tabla 136: Presupuesto total

6. Conclusiones y trabajos futuros

6.1 Conclusiones

Una vez finalizado el trabajo podemos concluir que se han cumplido los objetivos y requisitos establecidos al inicio.

Se ha desarrollado el proyecto de un sistema de comercio electrónico a nivel global, dotado de una alta disponibilidad y escalabilidad, apoyándonos en herramientas OpenSource y sistemas Cloud punteros e innovadores con un modelo de uso IaaS, el cual nos permite controlar los costes a la vez que expandirnos a cualquier rincón del mundo sin grandes esfuerzos ni inversiones.

6.2 Trabajos futuros

Como cualquier proyecto, se debe tener un planteamiento de desarrollo y aprendizaje continuo con el fin de estar continuamente innovando y mejorando nuestro sistema. Por ello pasamos a establecer una serie de posibles líneas de mejora:

- Implementación de nginx como servidor web en sustitución de Apache.

- Implementación de varias cachés, tanto a nivel de código como puede ser APC como a nivel de servidor (Varnish).

- Migración de S3FS a Elastic File System (actualmente en pruebas).

- Añadir réplica de lectura a RDS para la mejora de los backups

- Implantación de almacenamiento SSD con IOPS aprovisionadas.

- Contratación de instancias reservadas a un año. Representa un gran ahorro de costes.

- Implantación de Amazon Aurora en vez de MySQL.

7. Bibliografía y Referencias

7.1 Bibliografía

[1-bib]. Servidor Apache – Wikipedia

https://es.wikipedia.org/wiki/Servidor_HTTP_Apache [consultado Octubre 2015]

[2-bib]. Lenguaje de programación PHP – Wikipedia

<https://es.wikipedia.org/wiki/PHP> [consultado Octubre 2015]

[3-bib]. Base de datos – Wikipedia

https://es.wikipedia.org/wiki/Base_de_datos [consultado Octubre 2015]

[4-bib]. Sistema Gestor de Bases de Datos – Wikipedia

https://es.wikipedia.org/wiki/Sistema_de_Gesti%C3%B3n_de_Bases_de_Datos
[consultado Octubre 2015]

7.2 Referencias

[1]. Sistema gestor de contenidos (CMS)

“Introducción a los Sistemas de Gestión de Contenidos (CMS) de código abierto”
<http://mosaic.uoc.edu/2004/11/29/introduccion-a-los-sistemas-de-gestion-de-contenidos-cms-de-codigo-abierto/> [consultado Agosto 2015]

[2]. OpenSource

“Welcome to The Open Source Initiative | Open Source Initiative”
<http://opensource.org/> [consultado Agosto 2015]

[3]. SAAS (Software as a service)

“Qué es el Software como servicio (SaaS)”
<http://intelinode.com/1329/software-como-servicio-saas> [consultado Agosto 2015]

[4]. PASS (Platform as a Service)

“¿Qué es una Plataforma como Servicio (PaaS)? - Dos Ideas.”
<http://www.dosideas.com/noticias/actualidad/504-ique-es-una-plataforma-como-servicio-paas.html> [consultado Agosto 2015]

[5]. SEO

“¿Qué es el SEO o posicionamiento web? Guía Novatos | iSocialWeb”

<http://www.isocialweb.es/que-es-el-seo-o-posicionamiento-web-guia-novatos/>

[consultado Agosto 2015]

[6]. Comparación sistemas E-commerce

“6 Top Ecommerce Platforms for Do-It-Yourself Small Businesses | CIO”

<http://www.cio.com/article/2449485/e-commerce/e-commerce-6-top-ecommerce-platforms-for-do-it-yourself-small-businesses.html> [consultado Agosto 2015]

“The Best eCommerce Software of 2015 | Top Ten Reviews”

<http://ecommerce-software-review.toptenreviews.com/> [consultado Agosto 2015]

“Magento o Prestashop, ¿cuál elegir? | Sugerendo”

<http://www.sugerendo.com/blog/prestashop/magento-o-prestashop-cual-elegir/>

[consultado Agosto 2015]

“Magento 1.8 Usage Statistics”

<http://trends.builtwith.com/shop/Magento-1.8> [consultado Agosto 2015]

“PrestaShop Usage Statistics”

<http://trends.builtwith.com/shop/PrestaShop> [consultado Agosto 2015]

“Prestashop o Magento? Te ayudo a Decidirte? | ElblogdelSEO.com”

<http://www.elblogdelseo.com/prestashop-o-magento-te-ayudo-a-decidirte/>

[consultado Agosto 2015]

“Comparativa Magento vs. Prestashop, ¿Cuál es mejor? | Programador Freelance Web y Móvil”

<http://www.danielnavarroymas.com/comparativa-magento-vs-prestashop-cual-mejor/>

[consultado Agosto 2015]

“Shopify vs. Magento Community Edition Comparison”

<http://ecommerce-platforms.com/compare/shopify-vs-magento-community-edition-ecommerce-platform-comparison> [consultado Agosto 2015]

“Shopify vs Magento Customer Review - Alternative to Magento”

<https://www.shopify.com/compare/shopify-vs-magento> [consultado Agosto 2015]

“The pros and cons of Magento vs Shopify - Quora”

<https://www.quora.com/What-are-the-pros-and-cons-of-Magento-vs-Shopify>

[consultado Agosto 2015]

- [7]. Shopify
“Crear Tienda Online | Desarrollo de Tienda Virtual - Tu Tienda En Linea”
<https://es.shopify.com/> [consultado Agosto 2015]
- [8]. Bigcommerce
“Bigcommerce: Ecommerce Platform & Shopping Cart Software”
<https://www.bigcommerce.com/> [consultado Agosto 2015]
- [9]. Magento Go
“Magento GO, una opción sencilla y barata para tiendas Magento”
<http://www.desarrolloweb.com/articulos/magentogo-solucion-tienda-barata.html>
[consultado Agosto 2015]
- “Important Announcement about Magento Go and ProStores | Magento”
<http://magento.com/blog/magento-news/important-announcement-about-magento-go-and-prostores> [consultado Agosto 2015]
- [10]. Woocommerce
“WooCommerce - a free eCommerce toolkit for WordPress”
<http://www.woothemes.com/woocommerce/> [consultado Agosto 2015]
- “WooCommerce, el plugin de Wordpress para tiendas online”
<http://www.cobdc.net/programarilliure/woocommerce-el-plugin-de-wordpress-para-tiendas-online/> [consultado Agosto 2015]
- [11]. Prestashop
“Prestashop”
<https://www.prestashop.com/es/> [consultado Agosto 2015]
- [12]. Magento
“eCommerce Software & eCommerce Platform Solutions | Magento”
<http://magento.com/> [consultado Agosto 2015]
- [13]. Uso de Magento en grandes empresas
“Top 10 world brands using Magento - Magento Help and Support by MageWorx”
<http://support.mageworx.com/index.php?/News/NewsItem/View/36/top-10-world-brands-using-magento> [consultado Agosto 2015]
- [14]. Comparación Magento y Prestashop
“Magento vs Prestashop comparison of the best in e-commerce | ITX”
<http://itxdesign.com/magento-vs-prestashop/> [consultado Agosto 2015]
- [15]. Top 10k y 100k sitios web
“BuiltWith Web Technology Usage Statistics”
<http://trends.builtwith.com/> [consultado Agosto 2015]

- [16]. SQL Injection
"SQL Injection - OWASP"
https://www.owasp.org/index.php/SQL_Injection [consultado Agosto 2015]
- [17]. XSS
"Cross-site Scripting (XSS) - OWASP"
https://www.owasp.org/index.php/Cross-site_Scripting_XSS [consultado Agosto 2015]
- [18]. SSL / TLS
"Servidor SSL/TLS - FNMT"
<http://www.cert.fnmt.es/catalogo-de-servicios/certificados-electronicos/certificado-servidor> [consultado Agosto 2015]
- [19]. Backoffice
"Back office - Wikipedia, la enciclopedia libre"
https://es.wikipedia.org/wiki/Back_office [consultado Agosto 2015]
- [20]. CloudLinux
"CloudLinux OS"
<https://www.cloudlinux.com/> [consultado Agosto 2015]
- [21]. Cpanel/Plesk
"The Hosting Platform of Choice | cPanel, Inc."
<http://cpanel.com/> [consultado Agosto 2015]
- "Web Server Management Tools - Plesk 12"
<http://www.plesk.com/> [consultado Agosto 2015]
- [22]. Hosting
"Qué es un Hosting Web, tipos de Alojamiento y cuál elegir"
<http://www.ciudadano2cero.com/que-es-un-hosting-web-tipos-alojamiento-cual-elegir/> [consultado Agosto 2015]
- [23]. Servidor VPS
"Servidores Privados Virtuales (I): Qué son y para qué sirven"
<http://www.xatakaon.com/tecnologia-de-redes/servidores-privados-virtuales-i-que-son-y-para-que-sirven> [consultado Agosto 2015]
- [24]. Servidor dedicado
"¿Que son los Servidores Dedicados?"
<http://www.solingest.com/ayuda-faqs-solingest/servidores-dedicados> [consultado Agosto 2015]

- [25]. Sistemas Cloud
“¿Qué es cloud? - acens”
<https://www.acens.com/cloud/que-es-cloud/> [consultado Agosto 2015]
- [26]. vCore
“What is virtual CPU (vCPU)? - Definition from WhatIs.com”
<http://whatis.techtarget.com/definition/virtual-CPU-vCPU> [consultado Agosto 2015]
- [27]. RAM
“Memoria RAM, ¿cómo funciona, cuánta montar y tipos?”
<http://computadoras.about.com/od/conocer-mi-computadora/f/que-es-La-Memoria-Ram.htm> [consultado Agosto 2015]
- [28]. Nodo
“Nodo - EcuRed”
<http://www.ecured.cu/index.php/Nodo> [consultado Agosto 2015]
- [29]. Deployment/Despliegue de aplicaciones
“What is Software Deployment? | This Little Program Went to Market”
<http://www.godtlandsoftware.com/word-press/2011/03/26/what-is-software-deployment/> [consultado Agosto 2015]
- [30]. APC
“PHP: Introducci&on - Manual”
<http://www.php.net/manual/es/intro.apc.php> [consultado Septiembre 2015]
- [31]. Memcached
“memcached - a distributed memory object caching system”
<http://memcached.org/> [consultado Septiembre 2015]
- [32]. Redis
“Redis”
<http://redis.io/> [consultado Septiembre 2015]
- [33]. Varnish
“Home | Varnish Community”
<https://www.varnish-cache.org/> [consultado Septiembre 2015]
- [34]. NFS
“28.3. Network File System (NFS)”
<https://www.freebsd.org/doc/handbook/network-nfs.html> [consultado Octubre 2015]

- [35]. Cheff
“Chef | IT automation for speed and awesomeness | Chef”
<https://www.chef.io/chef/> [consultado Septiembre 2015]
- [36]. Puppet
“What is Puppet? | Puppet Labs”
<https://puppetlabs.com/puppet/what-is-puppet> [consultado Septiembre 2015]
- [37]. Jenkins
“Welcome to Jenkins CI! | Jenkins CI”
<https://jenkins-ci.org/> [consultado Septiembre 2015]
- [38]. MySQL
“MySQL :: The world's most popular open source database”
<https://www.mysql.com/> [consultado Septiembre 2015]
- [39]. InnoDB
“MySQL :: MySQL 5.0 Reference Manual :: 14.2 The InnoDB Storage Engine”
<http://dev.mysql.com/doc/refman/5.0/en/innodb-storage-engine.html> [consultado Septiembre 2015]
- [40]. ACID
“MySQL :: MySQL 5.6 Reference Manual :: 14.2.1 MySQL and the ACID Model”
<https://dev.mysql.com/doc/refman/5.6/en/mysql-acid.html> [consultado Septiembre 2015]
- [41]. Microsoft Windows
“Windows y Windows 10 - Microsoft”
<http://www.microsoft.com/es-es/windows> [consultado Septiembre 2015]
- [42]. Linux GPL
“GNU General Public License - Wikipedia, la enciclopedia libre”
https://es.wikipedia.org/wiki/GNU_General_Public_License [consultado Septiembre 2015]
- [43]. Debian
“Debian -- El sistema operativo universal”
<https://www.debian.org/index.es.html> [consultado Septiembre 2015]
- [44]. CentOS
“CentOS Project”
<https://www.centos.org/> [consultado Septiembre 2015]
- [45]. RedHat
“Spain | Red Hat”
<http://www.redhat.com/es/global/spain> [consultado Septiembre 2015]

- [46]. Gestor de paquetes Dpkg
“Ubuntu Manpage: dpkg - Gestor de paquetes Debian de medio nivel”
<http://manpages.ubuntu.com/manpages/dapper/es/man1/dpkg.1.html> [consultado Septiembre 2015]
- [47]. Gestor de actualizaciones APT
“Apt-Get”
<https://help.ubuntu.com/12.04/serverguide/apt-get.html> [consultado Septiembre 2015]
- [48]. Apache Software Foundation
“Welcome to The Apache Software Foundation!”
<http://www.apache.org/> [consultado Septiembre 2015]
- [49]. Licencia PHP
“PHP: License Information”
<http://php.net/license/> [consultado Septiembre 2015]
- [50]. Acuerdo de Nivel de Servicio o SLA
“¿Qué es un SLA? | SpanishPMO”
<http://spanishpmo.com/index.php/que-es-un-sla/> [consultado Septiembre 2015]
- [51]. IaaS
“Entendiendo la nube: el significado de SaaS, PaaS y IaaS”
<http://www.genbetadev.com/programacion-en-la-nube/entendiendo-la-nube-el-significado-de-saas-paas-y-iaas> [consultado Septiembre 2015]
- [52]. Google Cloud
“Google Cloud Computing, Hosting Services & Cloud Support — Google Cloud Platform”
<https://cloud.google.com/> [consultado Septiembre 2015]
- [53]. Microsoft Azure
“Microsoft Azure: plataforma y servicios de informática en la nube”
<https://azure.microsoft.com/es-es/> [consultado Septiembre 2015]
- [54]. Rackspace
“Rackspace: Managed Dedicated & Cloud Computing Services”
<https://www.rackspace.com/es-us> [consultado Septiembre 2015]
- [55]. Amazon Web Services
“AWS | Cloud Computing - Servicios de informática en la nube”
<https://aws.amazon.com/es/> [consultado Septiembre 2015]
- [56]. Xen
“The Xen Project, the powerful open source industry standard for virtualization.”
<http://www.xenproject.org/> [consultado Septiembre 2015]

- [57]. Paravirtualización (PV)
“Informática en nube: Clases de virtualización | TechNet Magazine”
<https://technet.microsoft.com/es-es/magazine/hh802393.aspx> [consultado Septiembre 2015]
- [58]. Virtualización completa (HVM)
“Informática en nube: Clases de virtualización | TechNet Magazine”
<https://technet.microsoft.com/es-es/magazine/hh802393.aspx> [consultado Septiembre 2015]
- [59]. Acuerdo de procesamiento de datos en AWS
“Noticias Jurídicas”
<http://noticias.juridicas.com/actual/veractual.php?id=4680> [consultado Septiembre 2015]

“LAS AUTORIDADES DE PROTECCIÓN DE DATOS DE LA UNIÓN EUROPEA APRUEBAN EL ACUERDO DE PROCESAMIENTO DE DATOS DE AMAZON WEB”
<http://www.anima-lopdp.com/nt-1-58/LAS-AUTORIDADES-DE-PROTECCION-DE-DATOS-DE-LA-UNION-EUROPEA-APRUEBAN-EL-ACUERDO-DE-PROCESAMIENTO-DE-DATOS-DE-AMAZON-WEB-> [consultado Septiembre 2015]

“AWS Cloud Compliance”
<https://aws.amazon.com/es/compliance/> [consultado Septiembre 2015]
- [61]. AMI (Amazon Machine Image)
“Amazon Machine Images (AMI) - Amazon Elastic Compute Cloud”
http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/AMIs.html
[consultado Septiembre 2015]
- [62]. EC2 (Amazon Elastic Compute Cloud)
“AWS | Elastic compute cloud (EC2) de capacidad modificable en la nube”
<https://aws.amazon.com/es/ec2/> [consultado Septiembre 2015]
- [63]. EBS (Elastic Block Storage)
“AWS | Elastic block store (EBS) para almacenamiento persistente”
<https://aws.amazon.com/es/ebs/> [consultado Septiembre 2015]
- [64]. RDS (Amazon Relational Database)
“AWS | Servicio de bases de datos relacionales (RDS)”
<https://aws.amazon.com/es/rds/> [consultado Septiembre 2015]
- [65]. Region
“Regions and Availability Zones - Amazon Elastic Compute Cloud”
http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/using-regions-availability-zones.html [consultado Septiembre 2015]

- [66]. Zona de disponibilidad
“Regions and Availability Zones - Amazon Elastic Compute Cloud”
http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/using-regions-availability-zones.html [consultado Septiembre 2015]
- [67]. Multi-AZ
“Implementaciones en zonas de disponibilidad múltiples (Multi-AZ) de Amazon RDS”
<https://aws.amazon.com/es/rds/details/multi-az/> [consultado Septiembre 2015]
- [68]. Failover
“IP failover: un alojamiento sin interrupción - OVH”
https://www.ovh.es/servidores_dedicados/ip_failover.xml [consultado Septiembre 2015]

“Failover and failback: Server Clusters (MSCS)”
[https://technet.microsoft.com/en-us/library/cc757139\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc757139(v=ws.10).aspx) [consultado Septiembre 2015]
- [69]. Failback
“What is Failback? - Definition from Techopedia ”
<https://www.techopedia.com/definition/27071/failback> [consultado Septiembre 2015]

“Failover and failback: Server Clusters (MSCS)”
[https://technet.microsoft.com/en-us/library/cc757139\(v=ws.10\).aspx](https://technet.microsoft.com/en-us/library/cc757139(v=ws.10).aspx) [consultado Septiembre 2015]
- [70]. CloudWatch
“Amazon CloudWatch – Servicios de supervisión de la red y la nube”
<https://aws.amazon.com/es/cloudwatch/> [consultado Septiembre 2015]
- [71]. Grupo de Seguridad
“Amazon EC2 Security Groups for Linux Instances - Amazon Elastic Compute Cloud”
http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/using-network-security.html [consultado Septiembre 2015]
- [72]. S3 (Simple Storage Service)
“AWS | Almacenamiento de datos seguro en la nube (S3)”
<https://aws.amazon.com/es/s3/> [consultado Septiembre 2015]
- [73]. CloudFront
“AWS | Servicio de entrega de contenidos y transferencia de datos”
<https://aws.amazon.com/es/cloudfront/> [consultado Septiembre 2015]
- [74]. Autoscaling
“AWS | Auto Scaling”
<https://aws.amazon.com/es/autoscaling/> [consultado Septiembre 2015]

- [75]. Elastic Load Balancer
“AWS | Elastic load balancing para la gestión de tráfico en aplicaciones”
<https://aws.amazon.com/es/elasticloadbalancing/> [consultado Septiembre 2015]
- [76]. Elastic IP
“Elastic IP Addresses - Amazon Elastic Compute Cloud”
http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/elastic-ip-addresses-eip.html [consultado Septiembre 2015]
- [77]. VPC (Virtual Private Cloud)
“AWS | Red virtual privada en la nube (VPC)”
<https://aws.amazon.com/es/vpc/> [consultado Septiembre 2015]
- [78]. VPC-Subnet
“Your VPC and Subnets - Amazon Virtual Private Cloud”
http://docs.aws.amazon.com/es_es/AmazonVPC/latest/UserGuide/VPC_Subnets.html
[consultado Septiembre 2015]
- [79]. IAM
“AWS Identity and Access Management (IAM) en la nube”
<https://aws.amazon.com/es/iam/> [consultado Septiembre 2015]
- [80]. Internet Gateway
“Internet Gateways - Amazon Virtual Private Cloud”
http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Internet_Gateway.html [consultado Septiembre 2015]
- [81]. Router
“Route Tables - Amazon Virtual Private Cloud”
http://docs.aws.amazon.com/AmazonVPC/latest/UserGuide/VPC_Route_Tables.html
[consultado Septiembre 2015]
- [82]. Route53
“AWS | Amazon Route 53 – Servidor de nombres de dominio – Servicio de DNS”
<https://aws.amazon.com/es/route53/> [consultado Septiembre 2015]
- [83]. DNS
“Los servidores DNS, usos, características y configuración”
<https://norfipc.com/internet/servidores-dns.html> [consultado Septiembre 2015]
- [84]. Réplica lectura en base de datos
“Capítulo 6. Replicación en MySQL”
<http://manuales.guebs.com/mysql-5.0/replication.html> [consultado Septiembre 2015]
- [85]. HAProxy
“HAProxy - The Reliable, High Performance TCP/HTTP Load Balancer”
<http://www.haproxy.org/> [consultado Septiembre 2015]

- [86]. CPD
“Centro de procesamiento de datos - Wikipedia, la enciclopedia libre”
https://es.wikipedia.org/wiki/Centro_de_procesamiento_de_datos [consultado Septiembre 2015]
- [87]. MariaDB
“Welcome to MariaDB! - MariaDB”
<https://mariadb.org/> [consultado Septiembre 2015]
- [88]. Percona
“Percona – The Database Performance Experts”
<https://www.percona.com/> [consultado Septiembre 2015]
- [89]. Sistema de Gestión de Bases de Datos Relacionales (SGDBR)
https://es.wikipedia.org/wiki/Sistema_de_Gesti%C3%B3n_de_Bases_de_Datos
[consultado Septiembre 2015]
- [90]. Sysadmin
“Administrador de sistemas - Wikipedia, la enciclopedia libre”
https://es.wikipedia.org/wiki/Administrador_de_sistemas [consultado Septiembre 2015]
- [91]. Virtualbox
“Oracle VM VirtualBox”
<https://www.virtualbox.org/> [consultado Septiembre 2015]
- [92]. Vagrant
“Vagrant”
<https://www.vagrantup.com/> [consultado Septiembre 2015]
- [93]. Puppuppet
“PuPHPet - Online GUI configurator for Puppet & Vagrant”
<https://puppet.com/> [consultado Septiembre 2015]
- [94]. Git
“Git”
<https://git-scm.com/> [consultado Septiembre 2015]
- [95]. PhpStorm
“PHP IDE :: JetBrains PhpStorm”
<https://www.jetbrains.com/phpstorm/> [consultado Septiembre 2015]
- [96]. IDE
“What Is an IDE? - Building Web Applications”
<http://webdesign.about.com/od/webprogramming/a/what-is-an-ide.htm> [consultado Septiembre 2015]

- [97]. Phpmyadmin
“PhpMyAdmin”
<https://www.phpmyadmin.net/> [consultado Septiembre 2015]
- [98]. Ley Orgánica de Protección de Datos de Carácter Personal
“BOE.es - Documento consolidado BOE-A-1999-23750”
<http://www.boe.es/buscar/act.php?id=BOE-A-1999-23750> [consultado Septiembre 2015]
- [99]. Safe-Harbor
“What is Safe Harbor? - Definition from WhatIs.com”
<http://searchcio.techtarget.com/definition/Safe-Harbor> [consultado Septiembre 2015]

“Safe Harbor - List”
<https://safeharbor.export.gov/list.aspx> [consultado Septiembre 2015]
- [100]. Simple Email Service (SES)
“AWS | Amazon Simple Email Service (SES) – Servicios de correo electrónico basados en la nube”
<https://aws.amazon.com/es/ses/> [consultado Septiembre 2015]
- [101]. VPN (Virtual Private Access)
“Qué es un VPN y por qué deberías usarlo”
<http://blogthinkbig.com/que-es-un-vpn/> [consultado Septiembre 2015]

“Red privada virtual - Wikipedia, la enciclopedia libre”
https://es.wikipedia.org/wiki/Red_privada_virtual [consultado Septiembre 2015]
- [102]. Bloque CIDR (Classless Inter-Domain Routing)
“3. CIDR - Redes locales y globales”
<https://sites.google.com/site/redeslocalesyglobales/6-arquitecturas-de-redes/6-arquitectura-tcp-ip/7-nivel-de-red/2-escasez-de-direcciones-ip-soluciones/3-cidr>
[consultado Septiembre 2015]

“¿Que es el CIDR? – Subnetting o subneteo Parte 3 | SoporteTI”
<https://blog.soporteti.net/teoria/que-es-el-cidr-subnetting-o-subneteo-parte-3/>
[consultado Septiembre 2015]
- [103]. LAMP
“Herramientas de trabajo para entorno LAMP | www.keopx.net”
<http://www.keopx.net/blog/herramientas-de-trabajo-para-entorno-lamp> [consultado Septiembre 2015]

“LAMP - Wikipedia, la enciclopedia libre”
<https://es.wikipedia.org/wiki/LAMP> [consultado Septiembre 2015]

- [104]. SSH
“Protocolo SSH”
<http://web.mit.edu/rhel-doc/3/rhel-rg-es-3/ch-ssh.html> [consultado Septiembre 2015]
- [105]. Instancias t2 de Amazon
“Instancias T2 EC2 de Amazon”
<https://aws.amazon.com/es/ec2/instance-types/t2/> [consultado Septiembre 2015]
- [106]. Servidor web Apache
“Welcome! - The Apache HTTP Server Project”
<http://httpd.apache.org/> [consultado Septiembre 2015]
- [107]. Intérprete de PHP
“¿Que es Interprete PHP? | Tecnoweb2.com - Software y Diseño Web Empresarial”
<http://tecnoweb2.com/que-es-interprete-php> [consultado Septiembre 2015]
- [108]. Módulos de multiprocesamiento de Apache
“Módulos de MultiProcesamiento (MPMs) - Servidor HTTP Apache”
<http://httpd.apache.org/docs/2.0/es/mpm.html> [consultado Septiembre 2015]

“Diferencias entre apache prefork, event y worker - Vozidea.com”
<http://www.vozidea.com/diferencias-entre-apache-prefork-event-worker> [consultado Septiembre 2015]
- [109]. Multiprocesamiento tipo Prefork (mpm-prefork)
“prefork - Servidor HTTP Apache”
<http://httpd.apache.org/docs/2.0/es/mod/prefork.html> [consultado Septiembre 2015]
- [110]. Multiprocesamiento tipo Worker (mpm-worker)
“worker - Servidor HTTP Apache”
<http://httpd.apache.org/docs/2.0/es/mod/worker.html> [consultado Septiembre 2015]
- [111]. Swap
“Swap (Español) - ArchWiki”
[https://wiki.archlinux.org/index.php/Swap_\(Espa%C3%B1ol\)](https://wiki.archlinux.org/index.php/Swap_(Espa%C3%B1ol)) [consultado Septiembre 2015]

“All about Linux swap space | Linux.com”
<https://www.linux.com/news/software/applications/8208-all-about-linux-swap-space>
[consultado Septiembre 2015]

- [112]. Out of Memory (OOM)
“Out of memory - Wikipedia, the free encyclopedia”
https://en.wikipedia.org/wiki/Out_of_memory [consultado Septiembre 2015]

“When Linux Runs Out of Memory - O'Reilly Media”
<http://www.linuxdevcenter.com/pub/a/linux/2006/11/30/linux-out-of-memory.html>
[consultado Septiembre 2015]
- [113]. Single Point of Failure (SPOF)
“What is single point of failure (SPOF)? - Definition from WhatIs.com”
<http://searchdatacenter.techtarget.com/definition/Single-point-of-failure-SPOF>
[consultado Septiembre 2015]

“What is a Single Point of Failure (SPOF)? - Definition from Techopedia”
<https://www.techopedia.com/definition/4351/single-point-of-failure-spoof>
[consultado Septiembre 2015]
- [114]. GlusterFS
“GlusterFS in AWS | Gluster Community Website”
<http://blog.gluster.org/category/aws-en/> [consultado Septiembre 2015]
- [115]. S3FS
“s3fs-fuse/s3fs-fuse · GitHub”
<https://github.com/s3fs-fuse/s3fs-fuse> [consultado Septiembre 2015]
- [116]. FUSE
“FUSE: Filesystem in Userspace”
<http://fuse.sourceforge.net/> [consultado Septiembre 2015]
- [117]. Redis
“Redis”
<http://redis.io/> [consultado Septiembre 2015]
- [118]. ElasticCache
“AWS | Amazon ElastiCache – Servicio de caché en memoria”
<https://aws.amazon.com/es/elasticache/> [consultado Septiembre 2015]
- [119]. Petición GET y PUT en S3
“Tutorial: CORS en S3 y CloudFront | Celingest Blog – Feel the Cloud”
<http://blog.celingest.com/2014/10/01/tutorial-cors-en-s3-y-cloudfront/> [consultado Septiembre 2015]

- [120]. CDN (Content Delivery Network)
“Content Delivery Network (CDN): Qué es, para qué sirve y por qué no rompe con la Neutralidad de la Red”
<http://www.xatakaon.com/tecnologia-de-redes/cdn-que-es-para-que-sirve-y-por-que-no-rompe-con-la-neutralidad-de-la-red>
[consultado Septiembre 2015]
- [121]. Cookie Stickiness
“Duration-Based Session Stickiness”
http://docs.aws.amazon.com/es_es/ElasticLoadBalancing/latest/DeveloperGuide/elb-sticky-sessions.html
[consultado Septiembre 2015]
- [122]. X-Forwarded-For
“Forwarded HTTP Extension”
<http://tools.ietf.org/html/rfc7239> [consultado Septiembre 2015]

“X-Forwarded-For”
http://docs.aws.amazon.com/es_es/ElasticLoadBalancing/latest/DeveloperGuide/x-forwarded-headers.html
[consultado Septiembre 2015]
- [123]. Ataques de denegación de servicio (DDoS)
“¿Qué es un ataque DDoS, cómo se para y sirve como medio de protesta?”
<http://www.genbeta.com/web/son-los-ataques-ddos-efectivos-como-medio-de-protesta>
- [124]. Más información seguridad ELB
“Interioridades, seguridad y solución de problemas en ELB | Feel the Cloud”
<http://blog.celingest.com/2014/01/31/interioridades-seguridad-solucion-problemas-elb/>
[consultado Septiembre 2015]

“Cloud, Big Data and Mobile”
<http://harish11g.blogspot.com.es/2012/07/aws-elastic-load-balancing-elb-amazon.html>
[consultado Septiembre 2015]
- [125]. Fallos de seguridad SSL

“Qué podemos hacer ante el mayor fallo de seguridad de la historia de Internet”
<http://www.xataka.com/seguridad/que-podemos-hacer-ante-el-mayor-fallo-de-seguridad-de-la-historia-de-internet>
[consultado Septiembre 2015]

“Heartbleed, otro fallo extremadamente grave en una librería SSL”
<http://www.genbeta.com/seguridad/heartbleed-otro-fallo-extremadamente-grave-en-una-libreria-ssl>
[consultado Septiembre 2015]

“5 Problemas con la seguridad de HTTPS y SSL en la web”

<http://rootear.com/seguridad/5-problemas-con-https-y-ssl>

[consultado Septiembre 2015]

“«Freak», Google y los fallos encriptados - ABC.es”

<http://www.abc.es/tecnologia/redes/20141015/abci-google-encipta-fallo-01410152035.html>

[consultado Septiembre 2015]

“«Freak», el nuevo fallo de seguridad que ha puesto en jaque a Apple y Google - ABC.es”

<http://www.abc.es/tecnologia/redes/20150305/abci-apple-google-freak-201503041916.html>

[consultado Septiembre 2015]

[126]. Autoridad de Certificación (CA)

“Autoridad de Certificación”

<http://www.anf.es/es/certificacion/pki-anf-ac/autoridad-de-certificacion.html>

[consultado Septiembre 2015]

[127]. IOPS

“¿Qué son los IOPS de un disco duro?”

<http://computadoras.about.com/od/conoce-tu-pc-discos-duros/a/Iops-De-Un-Disco-Duro.htm>

[consultado Septiembre 2015]

“Amazon EBS Performance Tips”

http://docs.aws.amazon.com/es_es/AWSEC2/latest/UserGuide/EBSPerformance.html

[consultado Septiembre 2015]

[128]. CNAME

“Ayuda de Administrador de Google Apps”

<https://support.google.com/a/answer/112037?hl=es>

[consultado Septiembre 2015]

[129]. Cabecera OPTIONS

“HTTP/1.1: Method Definitions”

<http://www.w3.org/Protocols/rfc2616/rfc2616-sec9.html>

[consultado Septiembre 2015]

“The HTTP OPTIONS method and potential for self-describing RESTful APIs”

<http://zacstewart.com/2012/04/14/http-options-method.html>

[consultado Septiembre 2015]

[130]. CORS

“HTTP access control (CORS)”

https://developer.mozilla.org/en-US/docs/Web/HTTP/Access_control_CORS

[consultado Septiembre 2015]

- [131]. XAMPP
“XAMPP for Apache Friends”
<https://www.apachefriends.org/es/index.html> [consultado Septiembre 2015]
- [132]. DigitalOcean
“Simple Cloud Infrastructure for Developers”
<https://www.digitalocean.com/> [consultado Septiembre 2015]
- [133]. RFC1945
“Network Working Group - Request for Comments: 1945”
<http://www.rfc-base.org/txt/rfc-1945.txt> [consultado Septiembre 2015]
- [134]. RFC2616
“Network Working Group - Request for Comments: 2616”
<http://www.rfc-base.org/txt/rfc-2616.txt> [consultado Septiembre 2015]
- [135]. RFC2774
“Network Working Group - Request for Comments: 2774”
<http://www.rfc-base.org/txt/rfc-2774.txt> [consultado Septiembre 2015]
- [136]. RFC2047
“Network Working Group - Request for Comments: 2047”
<http://www.rfc-base.org/txt/rfc-2047.txt> [consultado Septiembre 2015]
- [137]. Módulo “mod_remote_ip” Apache
“Modules | Directives | FAQ | Glossary | SitemapApache HTTP Server Version 2.5”
http://httpd.apache.org/docs/trunk/mod/mod_remoteip.html
[consultado Septiembre 2015]
- [138]. Archivo CSV
“CSV”
<https://es.wikipedia.org/wiki/CSV> [consultado Septiembre 2015]
- [139]. Linux Cron
“Cron & crontab, explicados”
<http://blog.desdelinux.net/cron-crontab-explicados/> [consultado Septiembre 2015]

“Command scheduling with cron”
https://www.debian-administration.org/article/56/Command_scheduling_with_cron
[consultado Septiembre 2015]
- [140]. Servidor SMTP
“Protocolos de mensajería (SMTP, POP3 e IMAP4)”
<http://es.ccm.net/contents/279-protocolos-de-mensajeria-smtp-pop3-e-imap4>
[consultado Septiembre 2015]

“Simple Mail Transfer Protocol”
https://es.wikipedia.org/wiki/Simple_Mail_Transfer_Protocol

[consultado Septiembre 2015]

[141]. AWS WorkMail

“Amazon WorkMail”

<https://aws.amazon.com/es/workmail/> [consultado Septiembre 2015]

[142]. MTA (Mail Transport Agent)

“Cómo funciona el correo electrónico (MTA, MDA, MUA)”

<http://es.ccm.net/contents/115-como-funciona-el-correo-electronico-mta-mdm-mua>

[consultado Septiembre 2015]

“Answers”

<http://www.networkredux.com/answers/linux-in-general/mail-and-mailservers/what-is-a-mailserver-or-mta>

[consultado Septiembre 2015]

[143]. Exim

“Exim Internet Mailer”

<http://www.exim.org/> [consultado Septiembre 2015]

[144]. Tablas de hashes

“Estructuras de datos dinámicas/Tablas Hash”

https://es.wikibooks.org/wiki/Estructuras_de_datos_din%C3%A1micas/Tablas_Hash

[consultado Septiembre 2015]

[145]. EAV (Entity Attribute Value)

“Flat Catalog”

http://merch.docs.magento.com/ee_docs/content/catalog/catalog-flat.html

[consultado Septiembre 2015]

“Magento y el modelo de datos EAV”

<http://www.danielnavarroymas.com/magento-y-el-modelo-de-datos-eav/>

[consultado Septiembre 2015]

[146]. OPC PHP

“OPcache”

<http://php.net/manual/en/book.opcache.php> [consultado Septiembre 2015]

[147]. APC PHP

“Caché Alternativo de PHP”

<http://php.net/manual/es/book.apc.php> [consultado Septiembre 2015]

[148]. Memcached

“Chat”

<http://memcached.org/>

[consultado Septiembre 2015]

- [149]. Imagen Modelo EAV
“Allan MacGregor, EAV Model – Editorial Safari Books”
<https://www.safaribooksonline.com/library/view/magento-php-developers/9781782163060/ch03s02.html>
[consultado Septiembre 2015]
- [150]. Capistrano
“A remote server automation and deployment tool written in Ruby.”
<http://capistranorb.com/> [consultado Septiembre 2015]
- [151]. CodeDeploy
“Introducción a AWS CodeDeploy”
<https://aws.amazon.com/es/codedeploy/> [consultado Septiembre 2015]
- [153]. Compresión Gzip
“Habilitar la compresión GZip | googlESimple”
<http://www.googleessimple.com/habilitar-compresion-gzip.html>
[consultado Septiembre 2015]
- [154]. Top
“Linux Command Reference: top - VPSLink Wiki”
http://wiki.vpslink.com/Linux_Command_Reference:_top
[consultado Septiembre 2015]
- [155]. Htop
“Display top CPU processes Using htopMonitoring Add comments”
<http://www.debianadmin.com/display-top-cpu-processes-using-htop.html>
[consultado Septiembre 2015]
- [156]. Sitemap
“Ayuda de Search Console”
<https://support.google.com/webmasters/answer/156184?hl=es>
[consultado Septiembre 2015]
- [157]. Minificar de archivos JS y CSS
“Minificar recursos (HTML, CSS y JavaScript)”
<https://developers.google.com/speed/docs/insights/MinifyResources?hl=es>
[consultado Septiembre 2015]
- [158]. Hipervisor en virtualización
“Informática en nube: Clases de virtualización | TechNet Magazine”
<https://technet.microsoft.com/es-es/magazine/hh802393.aspx>
[consultado Septiembre 2015]
- “¿Qué son los Hipervisores?”
<http://www.datakeeper.es/?p=716> [consultado Septiembre 2015]
- “Diferencias entre virtualización basada en hipervisor o en host”

<http://www.josemariagonzalez.es/2011/11/17/diferencias-virtualizacion-basada-hipervisor-host.html>

[consultado Septiembre 2015]

[159]. Gtmetrix

“Website Speed and Performance Optimization”

<https://gtmetrix.com/> [consultado Septiembre 2015]

[160]. WebPageTest

“Website Performance and Optimization Test”

<http://www.webpagetest.org/> [consultado Septiembre 2015]

[161]. Servidor Proxy

“Qué es un proxy”

<http://es.ccm.net/fag/2755-que-es-un-proxy> [consultado Septiembre 2015]

8. Licencias

Por último, en este apartado presentamos un cuadro resumen con las distintas plataformas usadas y sus respectivas licencias de uso:

| Licencias | | |
|-----------|------------------------------------|--|
| Sistema | Licencia | Enlace |
| Debian | GPL | http://www.gnu.org/licenses/gpl-3.0.en.html |
| Apache | Licencia Apache 2.0 | http://www.apache.org/licenses/LICENSE-2.0 |
| PHP | Licencia de PHP - Creative Commons | http://creativecommons.org/licenses/by/3.0/legalcode |
| Mysql | GPL | http://www.gnu.org/licenses/gpl-3.0.en.html http://www.mysql.com/about/legal/ |
| Magento | OSL 3 AFL 3 | https://opensource.org/licenses/OSL-3.0 https://opensource.org/licenses/AFL-3.0 |
| Redis | BSD | http://opensource.org/licenses/BSD-3-Clause |

Tabla 137: Tabla resumen licencias

